A MACHINE LEARNING BASED LANGUAGE MODEL TO IDENTIFY COMPROMISED USERS

by

Tien Duong Phan

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
April 2018

This thesis is dedicated to my parents

Table of Contents

List of Tables . . . . . **v**

List of Figures . . . . . **vi**

Abstract . . . . . **viii**

List of Abbreviations and Symbols Used . . . . . **ix**

Acknowledgements . . . . . **xi**

Chapter 1 Introduction **1**

Chapter 2 Literature Review **5**

2.1 Account Compromise and Impersonation 5
2.1.1 Some Recent Incidents 5
2.1.2 Account Compromise and Impersonation Implications 6
2.1.3 Mitigating Account Impersonation Attacks 7

2.2 Account Impersonation and Spam Detection Methods 7
2.2.1 IP Geolocation 8
2.2.2 URL Analysis 9
2.2.3 Account Profile/Social Network Analysis 10

2.3 Summary 12

Chapter 3 Methodology **13**

3.1 Approach 13

3.2 Algorithms Used 15
3.2.1 Skip-gram Technique 16
3.2.2 Artificial Neural Network 16
3.2.3 Other Machine Learning Algorithms 19

3.3 Datasets 25
3.3.1 Reuters Dataset 25
3.3.2 Enron Email Dataset 27
3.3.3 Twitter Dataset 29
3.3.4 Data Summary 30
3.3.5 Visualizing Enron and Twitter Dataset 30

3.4 Features 31
3.4.1 Word Embeddings 34
3.4.2 Text Features 36

3.5 Summary 41

Chapter 4 Results 42

4.1 Identifying Users 43
4.1.1 Accuracy Evaluations on Five Users 43
4.1.2 Accuracy Evaluations on Ten Users 44
4.1.3 False Alarm Rate 46
4.1.4 Summary 48

4.2 Further Digital Forensics 49
4.2.1 Word Cloud Representation Groups of Users 49
4.2.2 Word Clouds Distances Visualization 55

Chapter 5 Conclusion and Future Work 58

5.1 Conclusion 58

5.2 Future Works 59

Bibliography 61

List of Tables

3.1	Data summary	31
3.2	Information of selected users from the Enron dataset	38
3.3	Username of selected users from the Twitter dataset	38
4.1	Accuracy Evaluation	42
4.2	Information of five new users from the Enron dataset	44
4.3	Information of five new users from the Twitter dataset	45
4.4	False Positive Rate Evaluation	48
4.5	Top ten most frequent words in Employee and Trader group	55

List of Figures

1.1 Sampling of security incidents by attack type, time, and impact [1] 1
1.2 Percentage spam rate of emails by year [2] 2
1.3 Monthly active users of some online social networks [4] 3

3.1 Proposed system 14
3.2 Feed-forward neural network 17
3.3 One-hot encoding 18
3.4 A single perceptron's output with ReLU activation function . 19
3.5 SVM hyperplane [27] 20
3.6 SVM Optimal hyperplane [27] 21
3.7 Example of a news story in Reuters dataset 26
3.8 Example of an email in Enron dataset 28
3.9 Word clouds for Enron datasets 32
3.10 Word clouds for Twitter datasets 33
3.11 Extracting word embeddings 34
3.12 Visualization of word embeddings 35
3.13 Short text feature extraction 37
3.14 Visualization of emails and users in the Enron dataset 39
3.15 Visualization of tweets and users in the Twitter dataset 40

4.1 Summary of 10-fold cross validation results for 5 users 43
4.2 Summary of 10-fold cross validation results for 10 users 45
4.3 Summary of 10-fold cross validation performance drop 46
4.4 Confusion Matrix [40] 47
4.5 Word cloud of Employee group in Enron email dataset 50
4.6 Word cloud of Trader group in Enron email dataset 51

4.7 Word cloud of Emplyee and Trader groups in Enron email dataset 52
4.8 Word cloud of Manager group in Enron email dataset 53
4.9 Word cloud of Legal Department in Enron email dataset . . . 54
4.10 Word clouds distances visualization 56

Abstract

Identifying compromised accounts on online social networks that are used for phishing attacks or sending spam messages is still one of the most challenging problems of cyber security. In this research, the author explore an artificial neural network based language model to differentiate the writing styles of different users on short text messages. In doing so, the aim is to be able to identify compromised user accounts. The results obtained indicate that one can learn the language model on one dataset and can generalize it to different datasets with high accuracy and low false alarm rates without any modifications to the language model.

List of Abbreviations and Symbols Used

ANN	Artificial Neural Network
API	Application Program Interface
C&C	Command & Control
CPU	Central Processing Unit
CSV	Comma-separated Values
FP	False Positive
FPR	False Positive Rate
HBO	Home Box Office
ID	Identification
IP	Internet Protocol
LibSVM	A Library for Support Vector Machines
PCA	Principal component analysis
RCV1	Reuters Corpora Version 1
ReLU	Rectified Linear Unit
SD	Standard Deviation
SVM	Support Vector Machine
t-SNE	t-Distributed Stochastic Neighbor Embedding
TN	True Negative

URL	Uniform Resource Locator
XML	Extensible Markup Language
β	Weight vector in Support Vector Machine model
β_0	Bias in Support Vector Machine model
c	Number of words in the training context of Skip-gram model
m	Number of words in a short text
s	Number of chosen words in embedding space
$\theta = (j, t_m)$	A candidate which divides data into two subsets in tree based algorithm, where j is a feature and t_m is a threshold
v'_w	Output vectors of Skip-gram model
v_w	Input vectors of Skip-gram model
W	Size of the vocabulary in Skip-gram model

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor - Professor Nur Zincir-Heywood. Your continuous support, guidance, and encouragement since the first day of my study helped me overcome all the difficulties that I faced during my research. I always admire your kindness and knowledge. You are the person who is always full of energy in work. I could not have imagined having a better supervisor than you.

Secondly, I would like to express my deepest gratitude to my parents. You are always beside me since my first day in this world no matter what. I am proud of you, who spend all the life working hard and nurture their children with all of their heart. I wish you all the best, and live happily as you are.

Last but not least, I would like to thank all Network Information Management and Security (NIMS) Lab members, who are excellent, supportive, and also funny. I would also like to thank Dalhousie University, Nova Scotia Government, and Mitacs for their supports toward my study and this thesis research.

Chapter 1

Introduction

For decades, Internet and online services have become essential to any organization and individual. However, they also open new security threats. The rapid changes in information communication technologies result in identities to be no more restricted by physical identities, but also include virtual identities. One person may have multiple virtual identities or accounts in different organizations or online services. This opens a new land for sophisticated attacks, such as compromising user accounts [1]. Compromised accounts can be used for phishing attacks or spreading spam messages on the Internet, especially on online social networks. Fig. 1.1 illustrates the security threats growth from 2014 to 2016, where the size of the circles represent the estimated impacts of the incidents in terms of cost, and many of these high impact incidents were caused by compromised accounts.

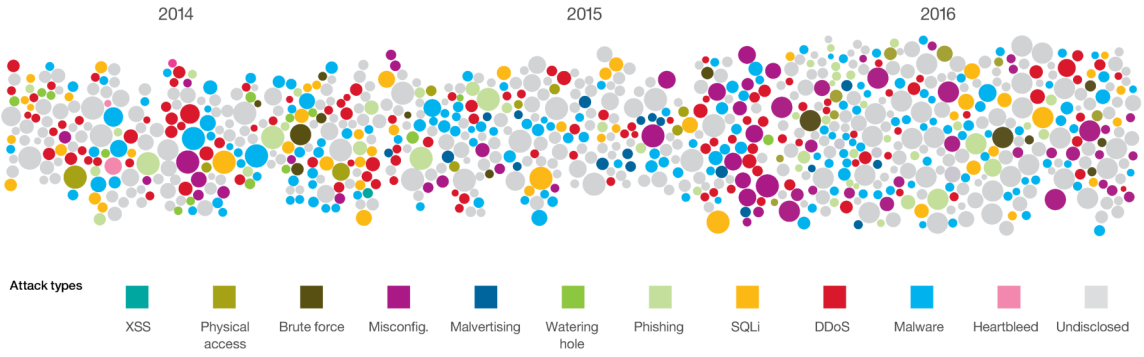

Figure 1.1: Sampling of security incidents by attack type, time, and impact [1]

The two main domains that this thesis focuses on are email and online social networks, which represent the working and daily life environments of users on the Internet, respectively. The latest security report carried out by Symantec [2], which was released in March, 2018, includes many cyber crime threat landscapes. It is evident that the email spam rate is always exceed 50% in the last 3 years and is

in an increasing trend as shown in Fig. 1.2. The spam rate is remarkable and is considered as a potential threat for security. Many of the spam campaigns [3] use compromised accounts to send spam emails to all the people in the victims' contact lists. Other statistic [4], as showed in Fig 1.3, compares the number of active users in different online social networks, and by Jun, 2017, there were 2 Billion Facebook users. These online social networks are great sources not only for marketing (such as users' interests in merchandises and services) and scientists (such as big data and data mining) but also cyber crime (such as spam spreading and information steeling).

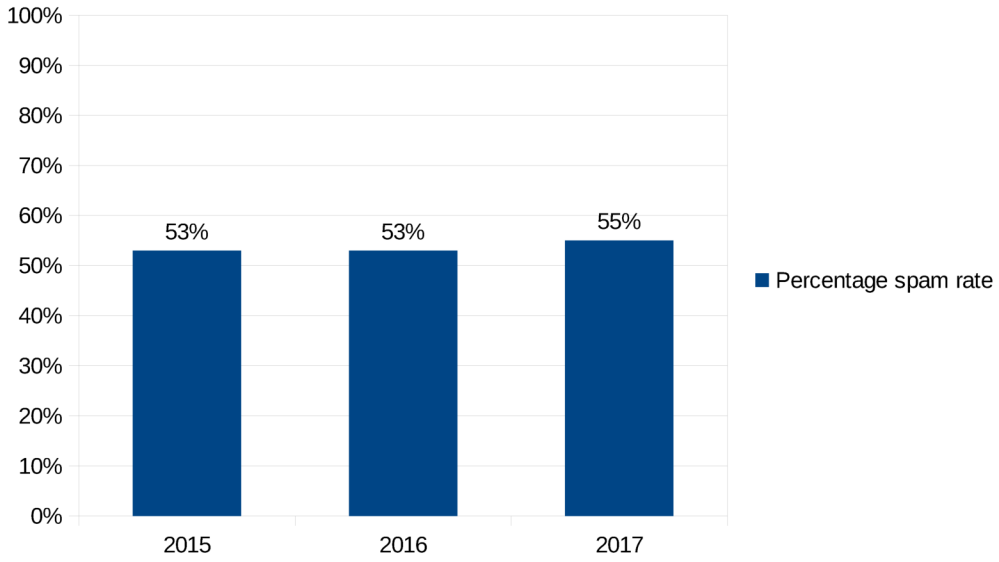

Figure 1.2: Percentage spam rate of emails by year [2]

Current online services recognize users by their registered identities, and such systems employ various authentication mechanisms to verify users as well as ensure that only legitimate users with registered identities are accessible to the services or able connect with other users in the networks. This very first stage not only contributes vitally to the security of the systems by determining the users logged into the system and the corresponding authorities, but also is a premise for user oriented services.

The assumption of the systems that only use authentication as a tool to protect users is that once the users are authenticated and gain their authorized access rights, they are considered as the proved identities until the end of the sessions. However, to

make use of this, attackers could either steal users' credentials or hijack to the current session. Well known techniques for these purposes are password stealing/cracking, cookie stealing, session hijacking, or even zero-day vulnerability exploits. This leads to many potential hazards such as online impersonation or information leakage.

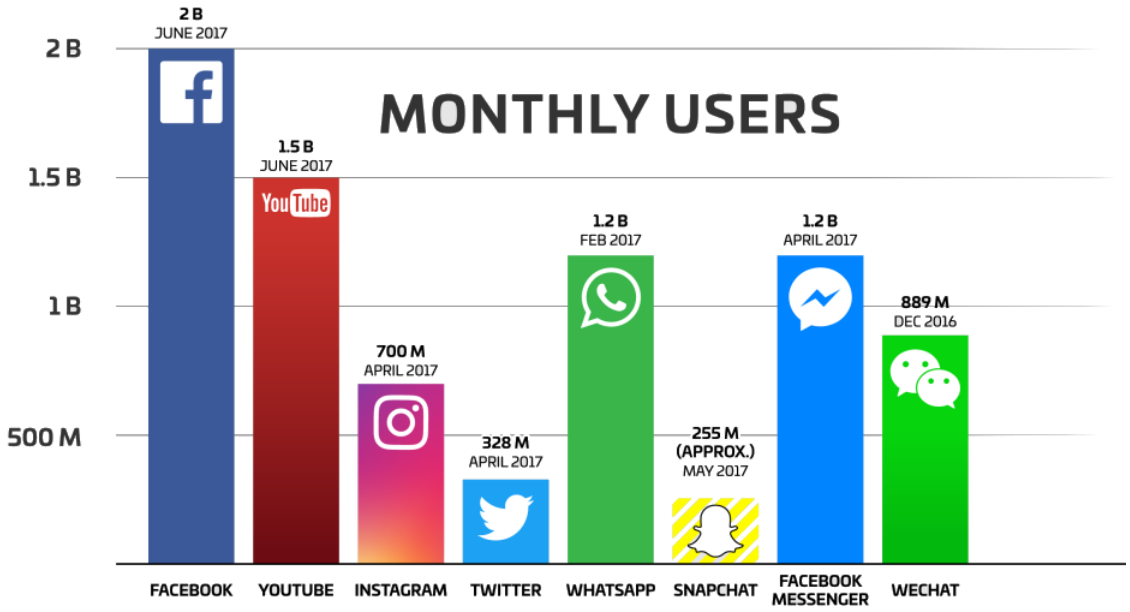

Figure 1.3: Monthly active users of some online social networks [4]

This thesis explores and evaluates the feasibility of an authorship attribution based approach [5] to identify compromised users for forensic analysis. To this end, a system which is based on artificial neural networks to perform an end to end digital investigation on users' short messages is implemented. One of the main contributions of the proposed system is that it can learn the language model from a typical dataset representing the language and can be applied to different types of short messages in different datasets.

In this work, the proposed system models the language on Reuters short news stories, which are generally one to two pages documents. The model is then evaluated for user identification on e-mail and tweet messages, which are even shorter than new stories. Evaluations and visualizations show that the proposed system is capable of modeling the language and can generalize this learned language model to identify different users based on their writing styles. Performance evaluations against other

learning systems show very promising results in the favor of the proposed system.

The rest of this thesis is organized as follows. Related work is reviewed in Chapter 2. The datasets, learning algorithms and the approach used in this research are discussed in Chapter 3. Evaluation and results are presented in Chapter 4. Finally, conclusions are drawn and the future work is discussed in Chapter 5.

Chapter 2

Literature Review

Online social networks have made a drastic change in our lives. Making friends and keeping in contact with them as well as their updates has become much easier. However, many downsides such as fake profiles or online impersonation have also grown. In this chapter, related works on compromised account and impersonation identification is reviewed. To this end, most of the recent studies focus on identifying spam or compromised accounts on online social networks. The following is a summary of these works to the best of the author's knowledge.

2.1 Account Compromise and Impersonation

Cyber crime are using account compromise and impersonation as the main method to spread spam and malware as well as extend their botnets. This section discusses some recent incidents that related to this trend. The large scale impacts of the incidents mention in this sections are the motivations for researchers to study and propose more sophisticated and efficient mitigation methods.

2.1.1 Some Recent Incidents

Well known social network accounts are tempting targets to hackers and cyber crime groups due to the fact that these accounts have large amount of followers and are great source to spread spam. Given the popularity of social networks, it is not hard to catch daily news of breaches that related to these services. For instant, Home Box Office (HBO) Twitter accounts were under attack in 2017 [6].

OurMine, the group of hackers responsible for the incident also involve hacking plenty celebrities' or famous groups' social network accounts. Some of the pioneers in technology are in the victim list, such as:

- Sundar Pichai - Google Chief Executive

- Mark Zuckerberg - Facebook Chief Executive
- Jimmy Wales - Wikipedia co-founder
- BuzzFeed
- TechCrunch

The victims in the incidents are those who have more understanding of technologies and security than regular users. This rises many concerns, such as whether or not users could protect their their data or even their privacy at all using online social networks. While waiting for the answer from service providers and researchers, most users accept to live with the problems as their connections are all included in these systems and a small number of others choose to quit the online communities to protect their privacy.

In the mentioned incidents, the accounts were directly targeted and hacked. However, there are even more sophisticated tricks that allow hackers to take over control of users' accounts without targeting the victim individually. One example is when the attackers gained access to Twitter Counter's service, and then freely sent tweets on behalf of many official Twitter accounts [7]. It is apparent that even large online social network's platforms may have deadly loopholes, and they are left unaware until cyber crime exploited them, harvested users' data and/or caused damages. These incidents emphasize the fact that everyone could be vulnerable to hackers by different ways, especially through online social networks.

The mentioned incidents are only some of the account compromise and impersonation incidents. The common point of the incidents is that they are only revealed after the attacks are completed. The reason is the lack of sophisticated mechanisms again such attacks, and as discussed earlier, current systems mainly relay on authentication mechanisms, which is insufficient.

2.1.2 Account Compromise and Impersonation Implications

Compromised online social accounts are becoming the popular tools for spreading spam and malware due to their effectiveness, and most of the spam campaigns employ compromised accounts to distribute spam [8] [9]. Exploiting the fact that users usually

read and respond to their friends promptly, social spam has higher distribution pace and successful rate than traditional email.

In 2010, Gao et al. [8] study on spam campaigns launched using compromised Facebook accounts. More than 187 million messages from approximately 3.5 million Facebook users' wall - Where Facebook users can create new posts or comment on existing posts - were collected and analyzed to quantify and characterize the spam campaigns. Notably, the study proofs that more than 97% of Facebook spam messages are originated in compromised accounts, rather than malicious accounts that are dedicated for spreading spam.

In the same year, Grier et al. [9] analyze spam on Twitter, and the research group observe that the accounts that send spam are legitimate accounts that have been compromised by spammers, and the successful rate (or click-through rate of spam URLs) are 0.13%, compared to considerably lower rates of email spam. The observation match the conclusion of Gao et al., and these studies both perform offline analyses on collected data.

2.1.3 Mitigating Account Impersonation Attacks

Mitigating online social network spam campaigns and account impersonation attacks are not always the simple problems to solve, and the reason is that compromised accounts are originally own by legitimate users, unlike dedicated spam accounts, which are created solely to serve malicious purposes. The solution need to include mechanisms to treat compromised account differently, and cannot just ban or remove these accounts upon detection due to potential negative impact to normal user experiences. For example, those accounts may still be actively used by their legitimate benign owners. Therefore, the security system must be able to detect potential compromised accounts before deciding mitigating method.

2.2 Account Impersonation and Spam Detection Methods

Recently, many solutions have been published against account impersonation and spam detection methods. The solution can be categorized into three main streams. This section discusses the approaches in more detail. The approaches are listed below:

- IP Geolocation
- URL Analysis
- Account Profile/Social Network Analysis

2.2.1 IP Geolocation

Online social networks are known to employ IP geolocation logging to battle against account compromise/impersonation. However, this approach is known to suffer from low detection rate. Two standout works that employ this method are discussed in this section.

In 2011, Stone-Gross et. al. [10] study the the Pushdo/Cutwail botnet spam campaigns in various aspects, including:

- Email address lists of victims
- IP addresses black lists
- The durable of bots

All the Command & Control (C&C) servers of the botnet are also listed and examined in this work. The authors analyzed the botnet's infrastructure and the spam spreading method to understand the working mechanism of the botnet. The study point out that new compromised computers and accounts are the most valuable bots in the botnets, since these bots are more likely able to pass though the IP back list filters. This is also the major disadvantage of IP geolocation logging method, since it can only block the detected C&C servers and can be easily bypassed by adding new C&C servers. Moreover, this method is unable to stop distributed botnets (i.e., the botnets that use Peer to Peer architecture instead of traditional client and server architecture) because the vast number of bots, not to mention that the bots' IP addresses cannot simply be blocked, which would affect the owner of the compromised devices or accounts.

In 2012, Thomas et.al. [11] study some interesting aspects of spam campaigns, including Spam-as-a-Service for political engagement. In order to achieve the goal, the authors examined the spam spreading mechanism and compromised accounts that were used in the campaigns. Their results suggested that the compromised accounts

are often in conjunction with hosts that are affected by malwares. This motivated the authors to study the connection between the purpose of the spam campaign and the related IP addresses (i.e., the IP address of the spam spreading sources as well as the bots). However, the work is more like an investigation on the already happened incidents than a contribution to detect and mitigate spam campaigns, since the authors do not propose any method to detect new compromised accounts/devices.

2.2.2 URL Analysis

The methods that belong to URL analysis family are based on the assumption that spam messages normally include URLs that link to other resources, which could be spam information or even malware, and analyzing the URLs can detect spam messages effectively. Some of the studies in this trend also consider distinguishing dedicated spam accounts (which are created by attackers for spamming purpose) and compromised accounts after spam messages are detected.

In 2010, Stringhini et. al. [12] established a set of "honey-profile" on three major online social networks to capture and analyze spam messages that were sent to the accounts. The scope of this study includes online social networks that are listed below:

- MySpace, which used to be the largest online social network at the time when the study was carried out.
- Twitter, one of the largest online social network in the world.
- Facebook, a giant technology company and social network.

The authors then proposed a method to detect spammers by analyzing the collected data. They also made an effort to connect spam messages with spam campaigns. The evaluations carried out showed that the proposed system was able to detect spammers by analyzing the received messages automatically. The main contribution of their study was the use of machine learning algorithms to differentiate spammer and legitimate users by analyzing the URLs in their messages. However, the data collection method seemed to collect more spam messages rather than normal messages, since the profiles were created to attract spam messages and not from normal users.

Therefore, the experiment and results are more controllable than the real online social network environment, which could lead to the considerably lower performance of the system in real world data.

In 2013, Egele et al. [13] proposed a tool, named COMPA, for compromised accounts detection using URL analysis. The experiments were performed on two major online social networks - Twitter and Facebook. The tool mainly examined the information that is related to the URLs that appear in the messages. Some important features employed in this study are listed below:

- URL entropy
- URL ratio
- URL repetition
- URL similarity

The proposed features seemed to work pretty well in the experiments carried out in their study. However, the tool was not able to deal with URLs that were shortened using URL Shortener Services, which could create multiple different URLs that point to the same "original URL". This weakness reduces the practicality of the proposed system considerably.

2.2.3 Account Profile/Social Network Analysis

Other existing approaches involve account profile analysis and social network analysis. These approaches based on the features that are related to users, such as, users' profile or connections.

In 2013, Thomas et. al. [14] conducted an account profile analysis against spam and abuse. The list of system under the analysis are listed below:

- Twitter, an online social network.
- Facebook, an online social network.
- Google, a popular search engine and many other service in conjunction.
- and Yahoo, which used to be a popular search engine and text messenger.

The motivation of their study was based on the assumption that automatically generated profiles/accounts were sold in large number by cyber crime to perform scams, phishing and spreading malware. Therefore, detecting and suspending/removing those accounts early could reduce potential security crisis. An investigation on black markets of Twitter accounts was carried out by the authors in 10 months. The investigation collected data from 270 merchants, such as:

- accs.biz
- victoryservices
- dataentryassistant.com

The collected data was used to build a fraudulent accounts detection system. The proposed method mainly covered the naming pattern that were normally used in fake accounts. The system was train on the usernames to generate regular expressions that could detect fake accounts. However, this approach works inefficiently with short names, which are more difficult to be recognized as the patterns. Moreover, the method is unable to detect compromised accounts that are abused to spread spams, due to the fact that the real users' profile are remain unchanged after being compromised.

In 2015, Zheng et. al. [15] studied social networks to detect and prevent spam. The authors analyzed the collected information determine the spam account. The features employed in this feature include:

- Mentions (i.e., @ $< username >$)
- Hashtags
- Number of connections with other users
- Users' researches

In this study, a large amount of users and text messages on Sina Weibo - a popular social media sites in China - are collected for examination. This approach seems to be more complicated and require to use more features, and the features are selected manually based on the authors' experience. This tends to ignore interesting data that could be spotted by machine learning algorithms.

2.3 Summary

As discussed in this chapter, the aforementioned approaches have their own disadvantages, such as:

- Account profile analysis tend to have lower accuracy, because their profiles are the original users' information which is likely to remain intact by spammers.
- Embedded URL analysis have the challenge of timely maintenance and update, since new spam URLs are generated/changed quickly.
- Social network analysis is complicated and features are selected manually.
- IP Geolocation fails to detect dedicated spam accounts and compromised accounts that are used for spamming purpose.

Therefore, instead of analyzing online social network features, embedded links, users' profile, or IP Geolocation, this thesis take a document authorship attribution based approach [5] on compromised accounts. This approach is enabled by the development of efficient natural language processing algorithms, and the main difference of this approach from the mentioned method is that it targets the text generated by users, rather than the location of users, user profiles, or embedded URLs. The approach also take the advantage of the ability to discover the important data automatically of machine learning algorithm to build generalizable systems on different type of texts without any modification.

Chapter 3

Methodology

In this chapter, a forensic analysis system which employs artificial neural networks to identify the users based on their writing styles is proposed. Specifically, short messages such as emails or tweets, where it is more difficult to identify whether the account is compromised or not [16], are analyzed.

3.1 Approach

Static and pre-designed modules such as list of key words or rule sets are not suitable against online impersonation, especially impersonation text messages. Hence, more adaptable and dynamic systems is needed to project such potential incidents. Machine learning algorithms are ideal candidates to cope with the problem because of their ability to extract learning features from large amount of data. The machine learning systems can be trained (supervised or un-supervised learning) and generalized to predict future data with high accuracy.

To this end, a shallow neural network for user identification on short text messages, which are used more frequent on online social networks, is trained. In doing so, this thesis aims to explore whether the text written is fraudulent or not, i.e., if the text is written by someone else, and not the actual account owner. The assumption of this approach is based on the idea that each person has his or her own writing style, and if we could analyze/learn on "enough" data generated by different users, we can distinguish different writing styles. In return, this could lead to a potential effective solution against detecting compromised accounts or impersonation. So whenever an account is taken over/compromised and used to send out spams or impersonation texts (i.e. when users' writing styles suddenly changed), the learning system could potentially flag the incidents. In the following, the learning algorithms, datasets and the features of the data that are used to design, implement and evaluate such a system will be discussed.

People can read and understand texts because we have been trained to understand the context of words, phrases, and sentences in the language we speak. Similarly, the computers/systems need to be firstly trained to "understand" the meanings/contexts of words. In order to do so, this thesis employs a simple yet effective technique, named Skip-gram [17], to learn the mapping from words to a numerical space. This technique elicits context of words from large text corpora in an unsupervised manner and results in similar words being grouped with each other. Note that we may not have enough data/context to learn the embedding of every words in the corpora, since there are words that only appear some times, so keeping the appropriate number of words to learn from the corpora is important. Output of the learning algorithm is a dictionary or encoder which encodes words into an embedding space. This enables similar words to be grouped together in the learned language model.

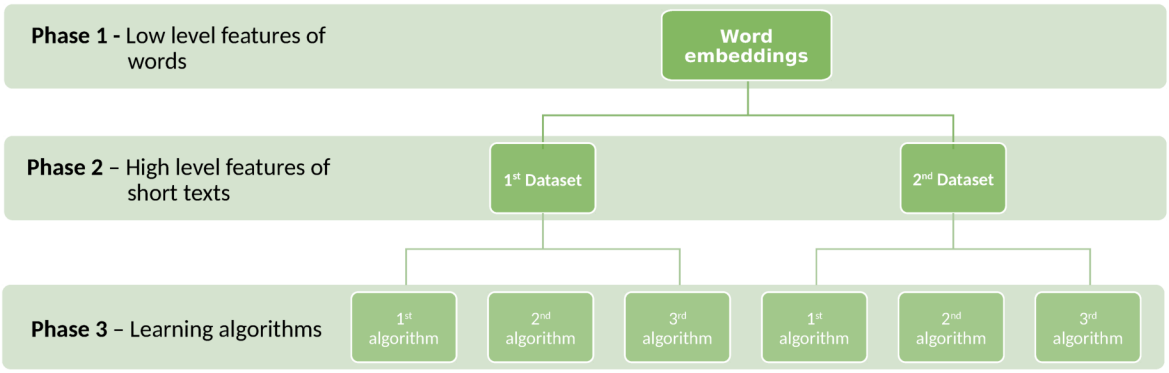

Figure 3.1: Proposed system

Fig. 3.1 illustrates the general approach of the proposed method, which includes three dependent phases. The phases must be performed in order and are listed as following:

1. The first phase is dedicated for learning the embedding of words from a large text corpora. The outputs are the word embeddings.

2. The second phase aims to produce higher level features of short texts in different datasets using the same low level features of word embeddings from phase 1. This work follows the bag of words approach to extract features of short texts. The average and standard deviation vectors, which are explained in detail in

section 3.4, are then computed from word embeddings of words that appear in the text.

3. The third phase uses outputs of the second phase as input for the machine learning algorithms to train them to classify short texts written by different users - in other words, to model their writing styles.

The size of short text features are fixed disregarding the length of the input short texts; Hence, the features are compatible to various machine learning algorithms. Due to this characteristic of the proposed approach, other datasets and algorithms could be added to the proposed system for evaluation. Note that the three phases are dependent, and it is important to have good word embeddings and short text features to prepare the input for the last phase. The decision maker in the last phase could be a artificial neural network, a decision tree, or any classification algorithm as discussed.

One of the advantages of this approach is that the first phase, which is the most time and resource consuming step, is only performed once as a preparation step. The language of the data is modeled in this very first phase. Then, this thesis explores whether the embeddings generated based on the model could be used to identify users for different datasets such as emails and tweets.

3.2 Algorithms Used

As discussed earlier, Skip-gram is employed for learning word embeddings and artificial neural network is used as the classifier for identifying the different users based on the word embeddings. Other well known classifiers including Naïve bayes, C5.0 Decision Tree and LibSVM are employed for comparison. This section discusses the algorithms and techniques that are used in this thesis, including:

- Skip-gram model
- Artificial neural network
- LibSVM
- C5.0

- Naïve bayes

3.2.1 Skip-gram Technique

Skip-gram model is designed and developed to learn word embeddings in a sentence or a document, in an unsupervised manner [17]. Formally, with any given sequence of training words $w_1, w_2, w_3, ..., w_T$, the Skip-gram model tries to maximize the average log probability as shown in Eq. (3.1):

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} log(p(w_{t+j}|w_t)) \tag{3.1}$$

where c is the number of words in the training context (which can be a function of the center word w_t). If a larger c is chosen, more training examples and probably higher accuracy will be created. However, the trade off is a longer training time. This study employs $c = 3$ (identified empirically) to save on training time. This value also worked well on all the datasets. Skip-gram defines $p(w_{t+j}|w_t)$ as shown in Eq. (3.2):

$$p(w_O|w_i) = \frac{exp({v'_{w_O}}^{\top} v_{w_I})}{\sum_{w=1}^{W} exp({v'_{w_O}}^{\top} v_{w_I})} \tag{3.2}$$

where v_w and v'_w are the input and output vectors that represent the word w, and W is the size of the vocabulary.

3.2.2 Artificial Neural Network

Artificial neural network is a machine learning algorithm where the neural networks are capable of learning in an unsupervised (supervised) fashion from data that is unlabeled (labeled). Artificial neural networks are popularly used in natural language processing, in research [18] [19] as well as in commercial products. This thesis follows the trend as it works on texts, which belong to a language, created by humans, and artificial neural network is a decision maker for classifying (identifying) the users based on the short text messages they wrote.

Artificial Neural Network Topology

In this study, the proposed model is a feed-forward neural networks [20] (all nodes are fully connected) with one hidden layer of 256 neurons. The Feed-forward neural

networks are also known as shallow neural network in some researches.

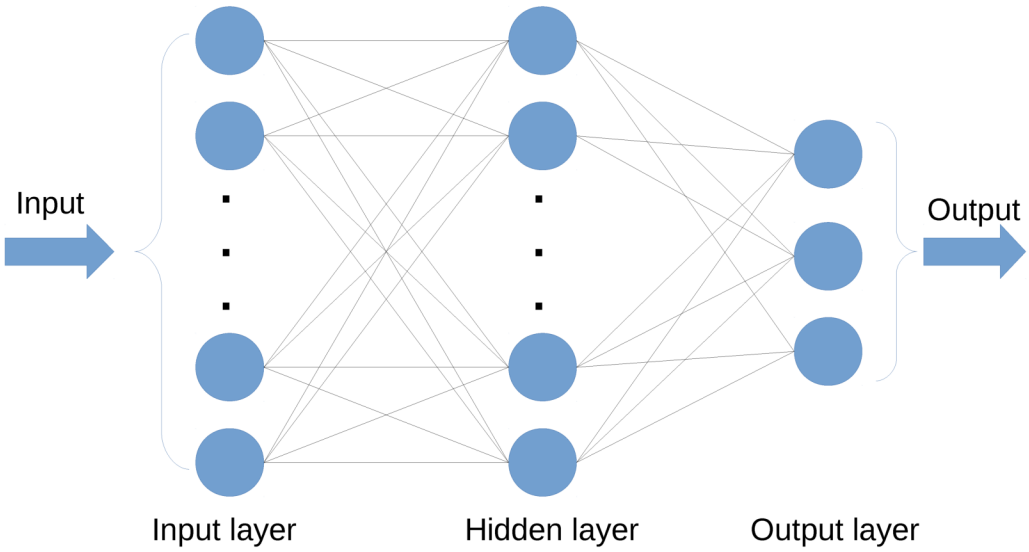

Figure 3.2: Feed-forward neural network

The neural network topology employed in this thesis is illustrated in the Fig. 3.2. Below is the list of characteristic of a feed-forward neural network:

- The perceptrons are arranged to form layers.
- The number of perceptrons in each layer and the number of layers are adjustable.
- All the layers between input and output layers are consider as hidden layers, since they do not have any connection with external world.
- Signals are constantly fed forward from the input layer, via the connections between layers, and output signals are produced at the output layer.

Note that there is no internal connection in the same layer, i.e, there is no connections between perceptrons within the same layer.

This thesis uses One-hot encoding to encode the classes; This results in the number of perceptrons in the output layer equals to the number of classes of data that are considering. This encoding is suitable for categorical data, such as the data that are

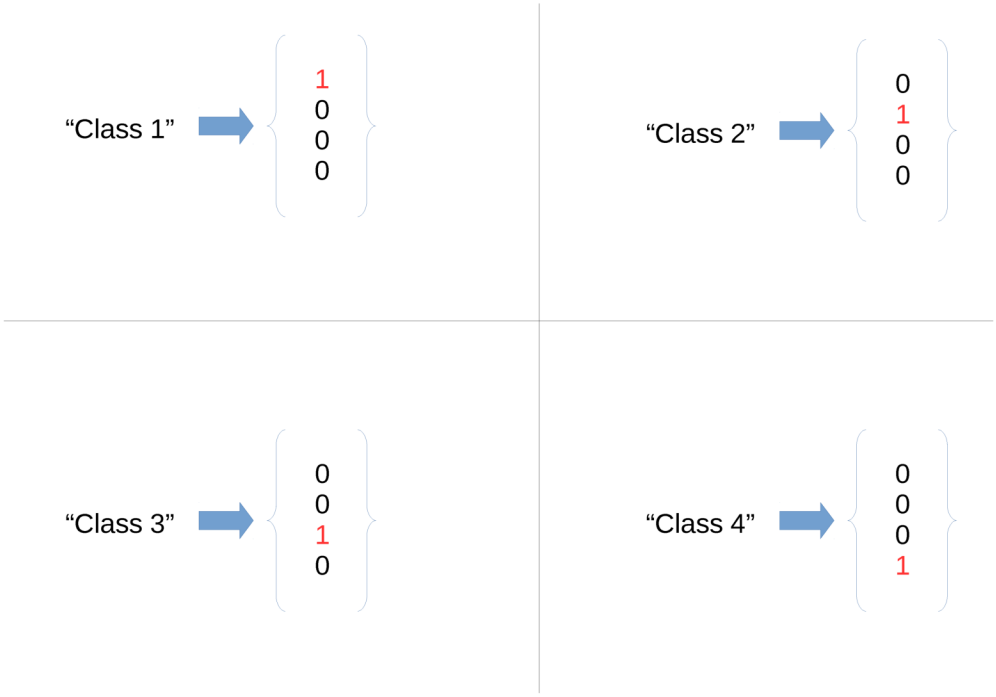

Figure 3.3: One-hot encoding

belonged to different users. As illustrated in Fig. 3.3, each class is encoded to an unique vector, and only one bit in the vector is activated, i.e., hot, and the other are in inactivated status.

Activation Function and Learning Algorithm

The thesis employs Rectified Linear Unit (ReLU) [21] - a non-linear activation function as the activation function of perceptrons. This activation is simple and being used widely in artificial neural networks, especially deep learning[22].

A perceptron that has ReLu activation function is illustrated in Fig. 3.4, where x_i is the i^{th} input and W_i is the corresponding weight associated with the input. The bias values that are associated with the perceptrons allows the ReLU activation function to be shifted to the left or right and able to produce better performance.

Adam optimization or Adam algorithm [23] is used to train the artificial neural network in this thesis. This optimization has effective step size and converge as quickly while require less fine tuning effort than other optimizer [24], such as Gradient Descent

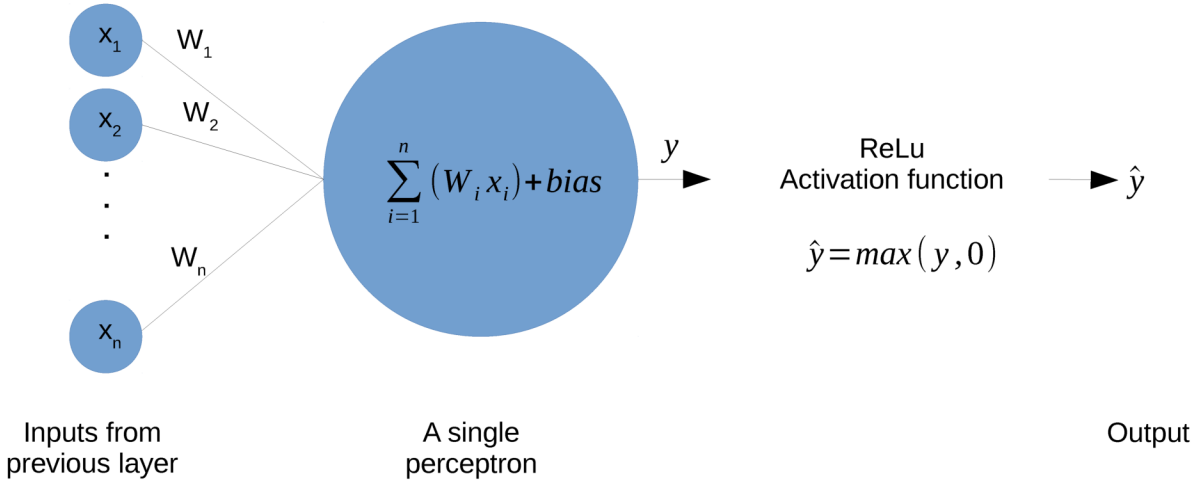

Figure 3.4: A single perceptron's output with ReLU activation function

Optimizer. This study keeps the hyper-parameter learning rate at 0.001, which is often used in training artificial neural networks.

3.2.3 Other Machine Learning Algorithms

All the machine learning algorithms which are listed and briefly discussed in this section are used to evaluate against the artificial neural network. The algorithms can be categorized into:

- Statistical classifiers (Naïve bayes)
- Decision tree (C5.0)
- Algorithm that uses hyperplane for classification purpose (Multi-class classification Support Vector Machine).

LibSVM

LibSVM [25] is a library for Support Vector Machines, which are supervised learning models for classification and regression analysis. Support Vector Machine (SVM) is useful and has been employed in document classification [26]. For classification

purpose, SVM is a discriminative classifier, which employs hyperplane to separate samples. The optimal hyperplane is determined in the training process (supervised learning), and used to classify new samples. Fig. 3.5 is a simple example of two classes that can be separated by a single line and visualized on a Cartesian Plane.

In the simple problem, there are more than one line that can separate the training samples of the two class with 100% accuracy. However, the optimal hyperplane (i.e., the optimal line in this case) is determined by the minimum distances to the samples in training set, and the line that has the largest distance is the optimal hyperplane. Through the definition of the optimal hyperplane, the algorithm tries to maximize the margins to the training samples, as visualized in the Fig. 3.6.

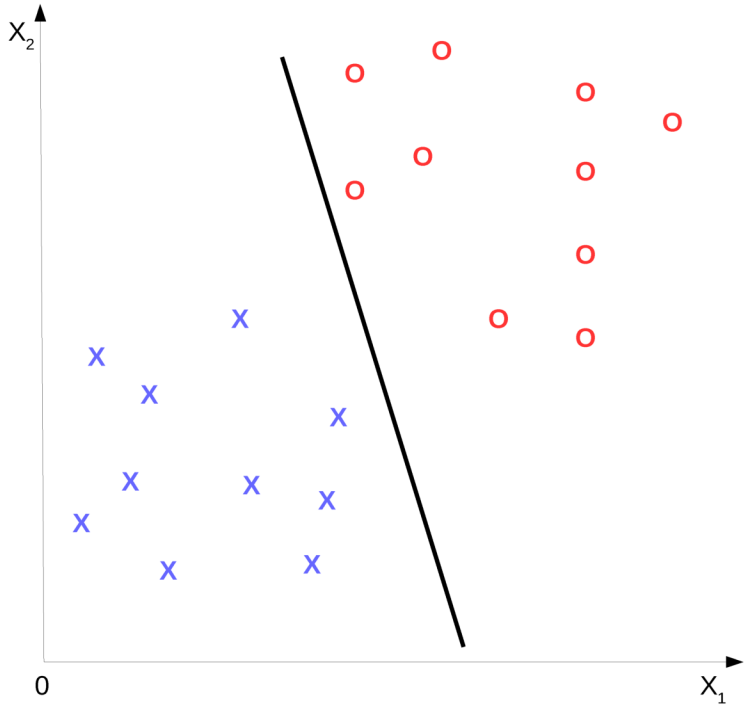

Figure 3.5: SVM hyperplane [27]

Formal definition of a hyperplane is showed in Eq. 3.3, where β is the weight vector and β_0 is the bias.

$$f(x) = \beta_0 + \beta^T x \tag{3.3}$$

By definition, there are infinite number of ways to scale β and β_0 that form optimal hyperplanes. One of the possible representation of optimal hyperplane that

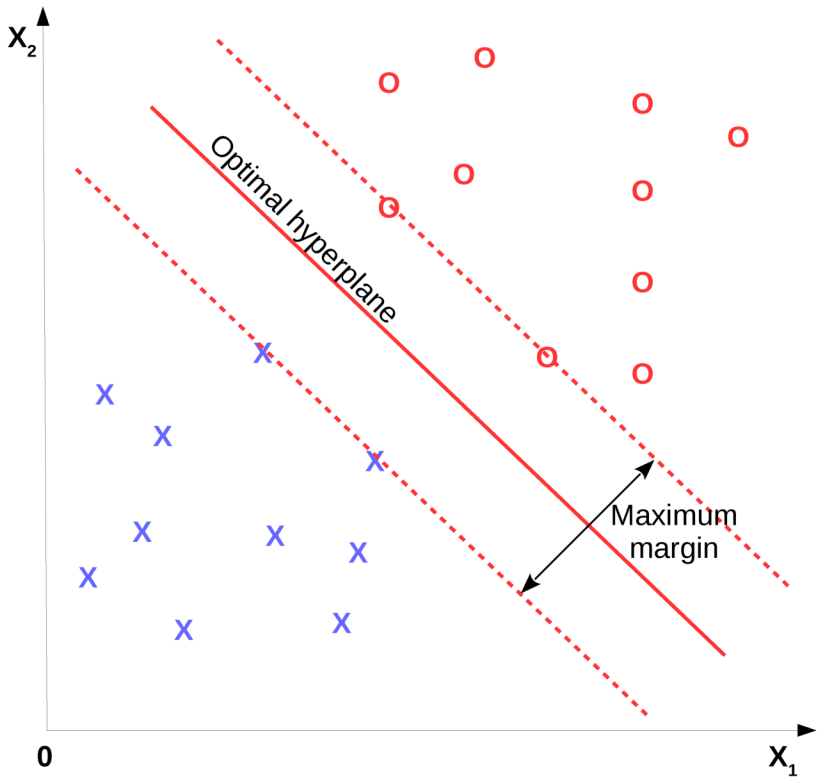

Figure 3.6: SVM Optimal hyperplane [27]

used widely is stated in Eq. 3.4, where x denotes the samples that are nearest to the hyperplane. These training samples are also known as support vectors.

$$|\beta_0 + \beta^T x| = 1 \tag{3.4}$$

In SVM models, the maximum margin is found based on the Eq. 3.5, where the aim is to minimize the function $L(\beta)$ which subject to the constraints. In the equation, y_i denotes labels of training samples, and the constraints require the hyperplane to differentiate all the training samples.

$$\min_{\beta,\beta_0} L(\beta) = \frac{1}{2}||\beta||^2 \; subject \; to \; y_i(\beta^T x_i + \beta_0) \geq 1 \forall i \tag{3.5}$$

‘ This work use the default setups of the library LibSVM, such as, type of SVM (C-SVC, which is used for multi-class classification). The type of kernel function employed is linear kernel function - the kernel that is commonly used in classification tasks with SVM.

C5.0

C5.0 algorithm [28] is an improvement version of C4.5, and the both algorithms are belong to tree based family, which use the divide-and-conquer technique. Tree based classification algorithms have many advantages as well as limitations [29]. Some of the major advantages of tree based classifiers are listed as following:

- Simple to visualize the model. As a result, the output of the algorithm is human friendly.
- Able to work with input features that is not normalized. This characteristic helps to reduce the complexity of pre-processing steps.
- Capable of learning from both numerical and categorical data.
- Employs a white box model, which make the explanation for the results easy for understanding. This characteristic is in contrast with black box models such as artificial neural networks.
- Require less resource, such as memory, CPU, and time. Therefore this algorithm have less issues working with large datasets.

- Output models can be validated using statistical tests, which can help validating the reliability of the model.
- Able to eliminate irrelevant feature in the feature set, and only relevant features are kept in the output decision trees.

Tree based classification models also have their limitations, such as:

- A small change in training data could change the output tree entirely, and the new trees could have different predictions on new samples.
- Overfiting on training data can create more complex trees. These threes perform well in the training data but are unable to keep the performance in test data.

Inputs for a tree based algorithm include a vector of features $X = [x_1, x_2, ..., x_n]$, where $x_i \in \mathbb{R}$, and a label vector $y \in \mathbb{R}^l$. In the Eq. 3.6, Q represents data at the note m. Each candidate $\theta = (j, t_m)$, where j is a feature and t_m is a threshold, divides the data into the defined Q_{left} and Q_{right} subsets.

$$\begin{aligned} Q_{left}(\theta) &= (x, y)|x_j \leq t_m \\ Q_{right}(\theta) &= Q \setminus Q_{left}(\theta) \end{aligned} \tag{3.6}$$

In the Eq. 3.7, the function $G(Q, \theta)$ is defined. The impurity metric at m is calculated by the function $H()$. Note that the function is dependent on the assigned task, which is classification or regression.

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{N_{right}}{N_m} H(Q_{right}(\theta)) \tag{3.7}$$

The goal of tree based algorithms is to minimize the impurity by using Eq. 3.8. The algorithm is performed recursively until it reaches the maximum allowable depth or an optimal solution is found.

$$\theta^* = argmin_\theta \; G(Q, \theta) \tag{3.8}$$

The implementation of C5.0 classification allows its output to be presented either as a decision tree or a rule set. In C5.0 models, samples are split based on the fields that provides the maximum information gain. C5.0's has some advantages over C4.5, such as higher speed, lower memory usage, smaller decision trees, and support for

boosting to increase accuracy. This thesis uses adaptive boosting option of C5.0, and this option generates ten classifiers rather than just one. Each classifier has it own vote, and the final class is determined by the most voted class.

Naïve Bayes

The Naïve Bayes Classifier [30] is based on Bayesian theorem and able to work with high dimension input features. Naïve Bayes is one of the most well known machine learning algorithm, which has been employed in numerous of studies since the middle of the 1990s.

Similar to any other machine learning algorithms, Naïve Bayes has its advantages and limitations [31]. The major advantages of the algorithms are:

- Has simple implementation thank to the simplicity of the Bayesian theorem.
- Able to handle discrete and continuous features, which gives the algorithm the ability to work with many different type of problems.
- Able to train the model with less data. This characteristic is useful when the data collection process is difficult and the available data is less than normal.
- Able to generate good results in most of the general problems.

Most of the limitations of Naïve Bayes come from the fact that the algorithm always assumes that the features are conditionally independence, which is reflected through the Bayes' theorem. The assumption is not always true, especially in complicated problems. The limitation lead to the fact that Naïve Bayes is unable to learn the interaction between features.

Bayes' theorem is stated in Eq. 3.9, where $x_1, .., x_n$ are independent features and y is a label.

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \tag{3.9}$$

Eq. 3.10 shows a simplified version of naive independence employed in Bayes' theorem.

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, ..., x_n)} \tag{3.10}$$

Finally, because $P(x_1, ..., x_n)$ is a constant (the values $x_1, ..., x_n$ are constants), we come up with the classification rule is stated in Eq. 3.11.

$$\hat{y} = \underset{y}{\text{argmax}}\ P(y) \prod_{i=1}^{n} P(x_i|y) \tag{3.11}$$

Despite its simplicity, Naïve Bayes algorithm is able to solve complicated tasks, such as, text classification [32]. Naïve Bayes is also a popular baseline algorithm for many classification problems, including text categorization. Therefore, Naïve Bayes is chosen as a baseline method in this work.

3.3 Datasets

This thesis uses three different datasets, including Reuters Corpora (RCV1) [33], Enron dataset [34], and Twitter dataset [35]. The RCV1 dataset is used for training the encoder (to learn the language) and the two remainder datasets are used to evaluate the proposed system on identifying different users based on the language learned. As discussed earlier, the main objective is to train the encoder once and then employ the encoder to analyze the writing styles of users in the new datasets. This section discusses the datasets that are used for training and evaluating the proposed system.

3.3.1 Reuters Dataset

In order to learn the embeddings of a language, i.e., English in this thesis, a large text corpora is needed as input for machine learning algorithms. This thesis uses the Reuters Corpus Volume I (RCV1) [33], which contains news stories in English. The dataset includes the articles written over a one year period, from 20^{th} August, 1996 to 19^{th} August, 1997.

The dataset contains approximately 810,000 Reuters news stories. It requires approximately 2.5 GB for storage of the uncompressed files, and the texts are stored in Extensible Markup Language (XML) format, which could be viewed using web-browsers or basic text editors. The XML files include different tags (or elements) to markup the news stories, such as:

- *<title>*

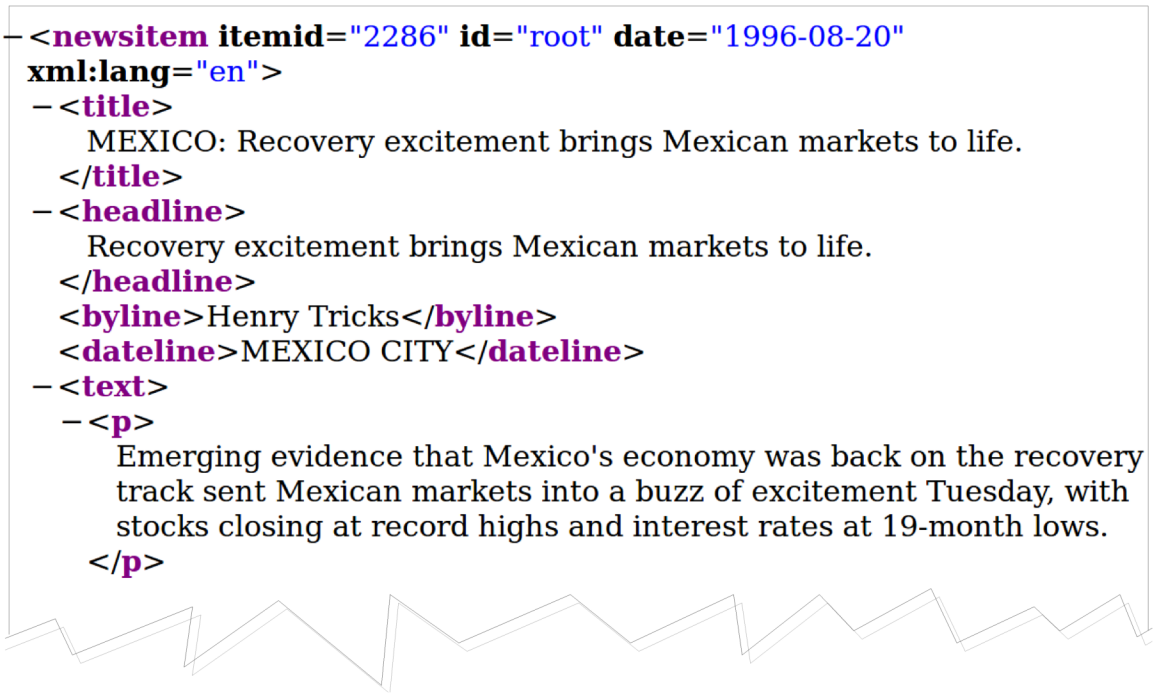

Figure 3.7: Example of a news story in Reuters dataset

- <*headline*>
- <*copyright*>
- <*metadata*>

Most of the content of the news stories are stored in the tag <*text*> (and its child tags), and the pre-processing step includes:

1. Extracting texts from the tag <*text*>.
2. Separating punctuation from words.
3. Standardizing the texts
    - Stripping multiple spaces and newlines.
    - Converting all word to lower-case letters.

The pre-processed data is then fed into the skip-gram model to learn and generate the word embedding vectors.

3.3.2 Enron Email Dataset

The Enron dataset [34] includes around 500,000 emails of 150 employees, and most of them are senior management of Enron corporation. Each email is a text file stored in a user's folder and/or sub-folders. The files include headers and the contents of the emails.

Fig. 3.8 shows an example of email in Enron dataset. In the figure, the header includes metadata of the email, such as:

- *Message-ID* that is indexed and used by email server.
- *Date*, which includes day and time that the email was sent.
- *From*, which includes email address of sender.
- *To*, which includes email dress of receiver(s).
- *Subject*, which includes the subject of the email.
- *X-cc*, which includes list of email addresses that receive carbon copy (if applicable).
- *X-bcc*, which includes list of email addresses that receive blind carbon copy (if applicable).

The main purpose of the pre-processing step is to extract only the texts that are written by users. The task includes the following steps:

1. Extracting the *From* and *To* fields of the input email:
    - Determining if the email is sent from the user using email address stored in the field *From*.
    - Determining if the email is sent to email addresses that are inside, outside of the Enron company by using the domain name in the field *To* in the header.
2. Extracting the contents that are composed by the user
    - Removing header of the emails.

```
Message-ID: <4396399.1075855680220.JavaMail.evans@thyme>
Date: Wed, 15 Nov 2000 08:10:00 -0800 (PST)
From: phillip.allen@enron.com
To: cbpres@austin.rr.com
Subject: Re: San Marcos Study
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Phillip K Allen
X-To: <cbpres@austin.rr.com>
X-cc:
X-bcc:
X-Folder: \Phillip_Allen_Dec2000\Notes Folders\Sent
X-Origin: Allen-P
X-FileName: pallen.nsf

George,

The other files opened fine, but I can't open winmail.dat files.
Can you resend this one in a pdf format.?

Thanks,

Phillip
```

Figure 3.8: Example of an email in Enron dataset

- Removing the forwarding content, since the content is not written by the user. Note that there are emails that only contain forwarded messages, and these emails are ignored.

3. Extract the email signature at the end of the email. The signatures are then used for determining the user's names and positions in the Enron company.

4. Separating punctuation from words.

5. Standardizing the texts, which includes stripping multiple spaces and newlines as well as converting all word to lower-case letters.

The emails of each user are then categorized into internal (sent to other Enron employees) or external emails (sent out of the company). In order to explore whether the model could learn the writing style of the users, the same number of internal and external emails are selected as samples of the users.

To minimize any potential biases based on employee groups, five employees are selected, including one manager, one trader, one from Legal Department, and two from regular employees. 2000 emails are then taken from each of the five users (1000 from internal emails and 1000 from external emails) - which sum up to 10,000 emails in total. The selected users and relevant information are discussed in the section 3.4.

3.3.3 Twitter Dataset

The Twitter dataset [35] includes tweets of 976 Twitter accounts between October 2014 and February 2015 collected by using Twitter API (Application Program Interface). The original research conducted with this dataset attempted to automatically link virtual and real-world identities based on information publicly available on the web using Twitter and Whitepages.com [36].

The tweets of each user are stored in a text file, and in comma-separated values (csv) format. Header of each file includes:

- *id*, which is the ID of the tweet.

- *created_at*, which it the date and time that the tweet was created/posted.

- *text*, which is the actual tweet content posted by the user.

- *Geo*, which includes geo-location of the user. This information can be easily harvested using Twitter API, surprisingly!

The pre-processing step in Twitter dataset is simpler than Enron dataset. The tweets composed by users are simply extracted from the column with the label *text*, and all the other data is simply ignored as they are not relevant to this work. The punctuation are then be separated from words, and texts are Standardized by tripping multiple spaces/newlines, and all letters are converted to lower-case.

In this research, five user accounts are randomly chosen, and 2000 tweets of each user are selected. The data taken is similar to Enron dataset. The aim of the choice is twofold:

- Firstly, to identify which tweets belong to which user account instead of linking the Whitepages identities with Twitter identities in the original paper of the dataset

- Secondly, to evaluate the proposed system in one more dataset (other than Enron email dataset) with the same setup.

3.3.4 Data Summary

Table 3.1 summarizes the data that is used to build the word embeddings and evaluate the proposed model. In short, the data used is described as following:

- Word embeddings are extracted from around 810,000 Reuters news stories to learn the language.

- Word embeddings are reused (i.e. use without any modification) in Enron and Twitter datasets with the same setup in term of amount of short texts and feature extraction method to explore if users can be identified/differentiated.

3.3.5 Visualizing Enron and Twitter Dataset

After the pre-processing step, the data is ready to be input for the proposed system. However, before any further steps are carried out, this study visualizes the obtained data to gain an insight into the data characteristics, especially the two datasets that are use for evaluating the proposed system.

Table 3.1: Data summary

Data used for word embeddings	810,000 Reuters news stories	
Data used in evaluations	**Users**	**Short texts/user**
Enron email dataset	5	2,000
Twitter dataset	5	2,000

Fig. 3.9 and Fig. 3.10 show the most frequent 200 words of Enron and Twitter datasets, respectively, in the form of word clouds. It is apparent that the words used in Enron dataset is more formal and related to the Enron corporation. Whereas the words used in the Twitter dataset are more informal and could be linked with the events happen in the users' daily lives. The choice of these datasets aims to explore whether the proposed approach is sensitive to certain types of data and written styles or not.

Through all the discussions and the steps that are performed, the pre-processed data are:

- Suitable for learning the Embeddings of words (data from RCV1 dataset).

- Written by users and has different in writing style (data from Enron and Twitter datasets).

3.4 Features

This research explores whether word embeddings could be used as features to differentiate different users of a given dataset with good accuracy. To this end, the embedding space of the most frequent 100,000 words are firstly learned from Reuters Corpora (RCV1). Then the single word features are used to form the text features in two different datasets to evaluate how well these features work. Details of the feature extraction methods are discussed in this section.

Figure 3.9: Word clouds for Enron datasets

Figure 3.10: Word clouds for Twitter datasets

3.4.1 Word Embeddings

Feature Extraction

A word embedding is a mapping form word space to a vector space of real numbers as shown in Eq.(3.12):

$$W : Words \rightarrow \mathbb{R}^n \tag{3.12}$$

This function maps words in a language to high-dimensional vectors as illustrated in Fig. 3.11, where $E(W_i)$ is embedding of i^{th} word ($1 \leq i \leq s$), and s is the number of chosen words. The distributed representations of words help improving the performance of learning algorithms by grouping words that have similar contexts.

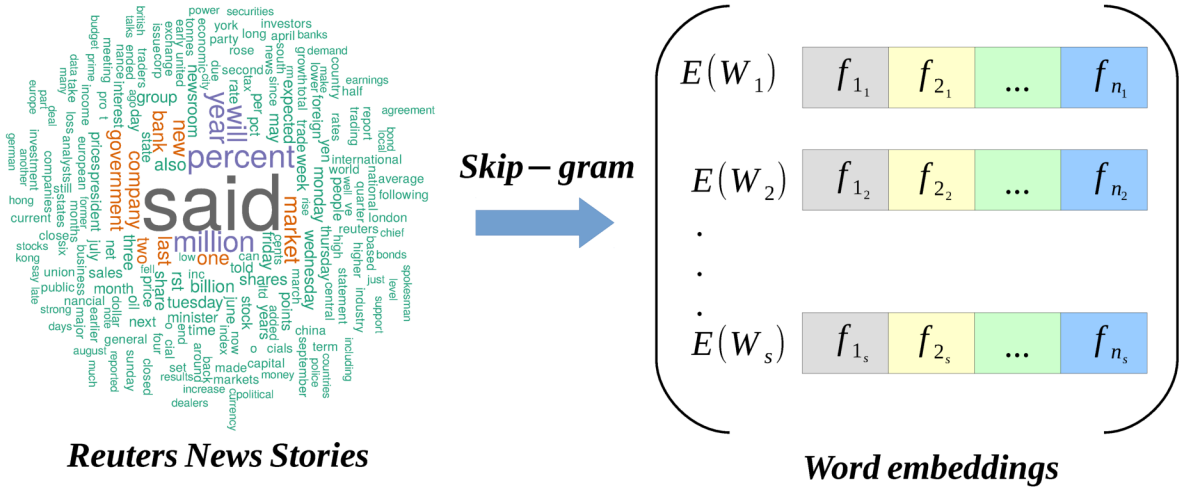

Figure 3.11: Extracting word embeddings

In this thesis, the word embeddings are 128-dimensional (chosen empirically) vectors of real numbers. The preliminary experiments show that this provides an efficient and effective data representation in terms of computational cost and accuracy trade-off. Note that the word embedding is learned from the RCV1 and represent general context of the most frequent 100,000 words in the corpus. Hence, the goal is to reuse these word embeddings on different datasets to capture the writing styles of different users.

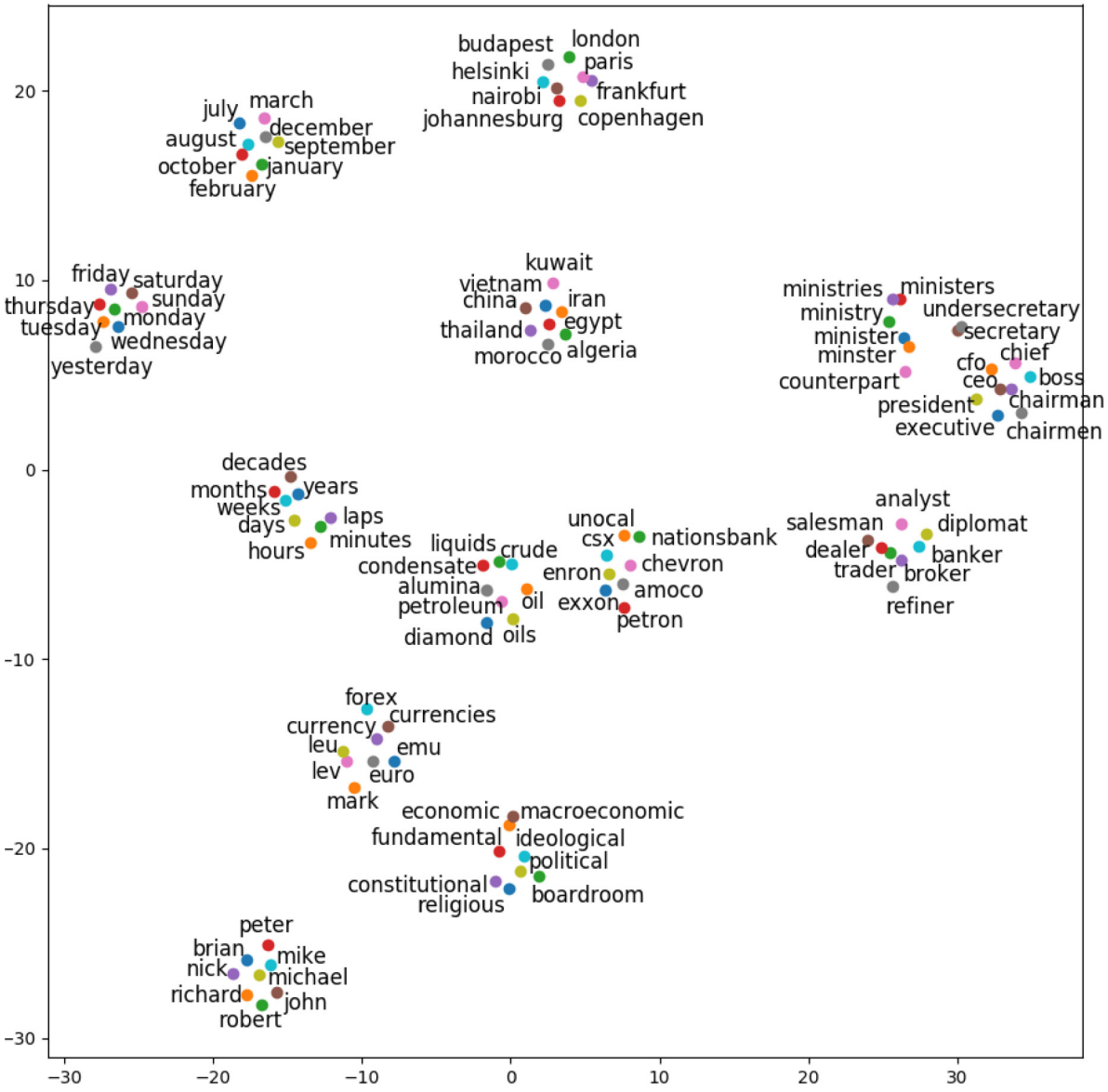

Figure 3.12: Visualization of word embeddings

Feature Visualization

Fig. 3.12 is a visualization for demonstrating the efficiency of the trained word embedding by querying the seven nearest words of the given words in the embedding space. The figure is obtained by using t-SNE [37] - a tool to visualize high-dimensional data which is integrated in scikit-learn library [38]. The list of chosen words include:

- *Monday*, *days*, and *March*, which represent time.
- *chairman*, *trader*, and *Minster*, which represent positions/jobs.
- *political*, which represents adjectives.
- *oil* and *currency*, which are related to economy.
- *London*, which represents city names.
- *Vietnam*, which represents countries.
- *Michael*, which represents peoples' names.
- *Enron*, which represents companies names.

It can be observed from the graph that days of a week, period of time, job positions, nouns, people's names, cities, and countries are grouped into separated groups.

Moreover, the groups of words which have similar context are close to each other. For example, the clusters that represent time including the days of the week, months of the year, and periods of time, are neighbors in the visualization. In this visualization, neighbor groups of the name *Michael* and *Enron* represent the names of people and name of companies, respectively. This is a proof that the embedding is not only work with regular words but also special words such as human or company names. Note that the entire embeddings are not plotted due to the density and interfusion of the vectors.

3.4.2 Text Features

Feature Extraction

Text features are high level features that are built on top of word embeddings, and this high level features are extracted from short text written by users. The general

steps to extract text features are listed below:

- Each word ($W_i, where\ 1 \leq i \leq m$) in the sort text is looked up and convert to embedding vector (i.e., word feature).

- Short text features are extracted on top of word features. As discussed earlier, short texts are treated as bags of words, and text features are the concatenation of the vector of means and standard deviations of word embeddings as demonstrated in Fig 3.13, where $\bar{f}_i$ and S_{f_i} are defined in Eq. (3.13) and Eq. (3.14), respectively.

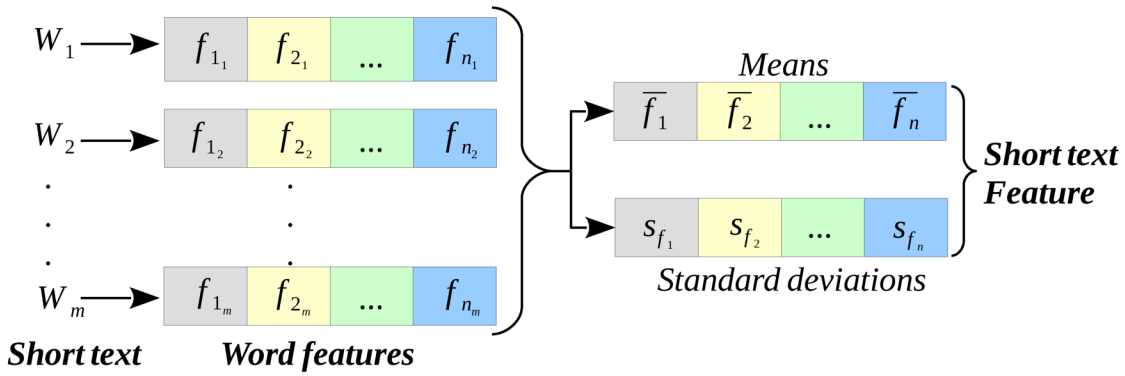

Figure 3.13: Short text feature extraction

$$\bar{f}_i = \frac{1}{m}\sum_{j}^{m}(f_{i_j}) \tag{3.13}$$

$$S_{f_i} = \sqrt{\frac{\sum_{j=1}^{m}(f_{i_j} - \bar{f}_i)^2}{m-1}} \tag{3.14}$$

Note that m is the number of words in the short text (email or tweet) represented. This results in 256-dimensional vectors (double the size of word embeddings) which contain the information of all used words (of an e-mail or a tweet) and the amount of variation or dispersion. In doing so, this thesis aims to capture and represent the user's writing style. The evaluation results (See Chapter 4) show that the proposed system is able to support the identification of users with high accuracy.

Table 3.2: Information of selected users from the Enron dataset

Username	**Position**
mann-k	*Employee*
germany-c	*Employee*
kaminski-v	*Manager*
bass-e	*Trader*
perlingiere-d	*Legal Department*

Table 3.3: Username of selected users from the Twitter dataset

Username
Ashley_Nunn75
bradshaw1984
shawnevans81
terrymarvin63
WhieRose65

Feature and User Visualization

This work also tries to link the job positions of users with their writing style in the visualization of the short text features. The extra information could potentially give us an insight into the difference in writing style of groups of users who have similar jobs when the users' texts are visualized. This especially useful for digital forensic, where the visualization and summary of data is vital to give hints and speed up the progress. However, this thesis can only study this relationship in the Enron email dataset, since the job of users are not given in the Twitter dataset. Table 3.2 and 3.3 shows the information of five chosen users from the Enron dataset (including usernames and jobs) and Twitter dataset (only usernames are listed), respectively.

One of the benefits of the feature vectors is that they enable us to visualize the data in 3-dimensional space by using Principal component analysis (PCA). Furthermore, this is independent of the classification algorithms used. The visualizations help us see the short text clusters, give an insight into the data characteristics.

Fig. 3.14 and Fig. 3.15 show the visualizations of the text features extracted from Enron email and Twitter datasets, respectively. In the figures, each point represents a short text and each of the five colors represents texts written by one user.

Figure 3.14: Visualization of emails and users in the Enron dataset

Figure 3.15: Visualization of tweets and users in the Twitter dataset

The clusters of colors, which include the points carry the same color, are able to spotted. However, the clusters are not trivial to be separated because of the density and interfusion. Interestingly, in the Fig. 3.14, the pink cluster, which is quite isolated from the others, includes the emails of the user who worked in Enron Legal Department. The distance of this cluster to the others seems to indicate that this user has a very different writing style which may be because of his/her position in the company. This is an example of an human friendly tool for digital forensic.

3.5 Summary

This chapter discusses all the ideas from the general approach to details of materials that are needed to build a machine learning model, including:

- Data that is used for building language model, training, and test the proposed system.
- Algorithms that are employed in this study.
- Feature extraction methods that employed to prepare inputs for machine learning algorithms.
- The visualizations along the way to get an insight into the data and features.

The discussions and visualizations in this chapter are also an archive of ideas to inspire the author himself in the future works related to digital forensics.

Chapter 4

Results

This chapter presents the results of the proposed system in term of detecting / differentiating users' writing styles. Successfully on detecting user writing style could lead to a potential effective solution against detecting compromised accounts or impersonation as discussed in previous chapters. This chapter also discuss the forensic application of the proposed system to strengthen the approach assumption of this thesis.

Table 4.1: Accuracy Evaluation

Algorithm		**Enron Email Dataset**		**Twitter Dataset**	
		5 users	**10 users**	**5 users**	**10 users**
ANN[a,b]	*Training*	94.12%	93.36%	92.68%	91.30%
	Test	82.79%	73.91%	81.64%	73.13%
	X-Validation[d]	85.08%	76.05%	84.14%	75.65%
C5.0 (Boosted)	*Training*	99.40%	99.10%	98.50%	99.20%
	Test	72.10%	60.80%	79.30%	65.00%
	X-Validation	76.90%	65.50%	79.90%	66.30%
libSVM[c]	*Training*	75.30%	57.74%	80.10%	75.79%
	Test	72.87%	50.06%	77.63%	69.40%
	X-Validation	75.16%	56.83%	80.03%	74.16%
Naïve Bayes	*Training*	43.30%	32.39%	53.26%	54.09%
	Test	40.53%	32.60%	51.97%	55.69%
	X-Validation	41.86%	32.14%	53.39%	54.32%

[a]Artificial Neural Network
[b]Average of 20 runs
[c]Linear Kernel
[d]10 Folds Cross Validation.

4.1 Identifying Users

This section reports and discusses the learning results of artificial neural network and other machine learning algorithms. The different algorithms are trained an tested on short text features, which are dissed in Chapter 3, in a supervised learning fashion, i.e., with labels given. Table 4.1 shows the results of classifying users of two different datasets. The results include accuracy on training (the first 70% of data) and test (the remaining 30% of data) partitions of the datasets using four different algorithms as shown. These results are analyzed in the Section 4.1.1 and 4.1.2. In order to minimize any biases that may exist in the data, 10-fold cross validation is also performed in all the algorithms.

4.1.1 Accuracy Evaluations on Five Users

The five users selected from The Enron and Twitter datasets are listed and discussed in Chapter 3. In terms of the performance on the test partitions, shallow neural network can accurately classify 82.79% and 81.64% short text written by different users in Enron email and Twitter datasets, respectively. It out performs boosted C5.0 decision tree, LibSVM and Naïve Bayes algorithms. However, it should be noted here that the task seems to be too difficult for Naïve bayes as the training and test accuracies are below 50% in half of the cases.

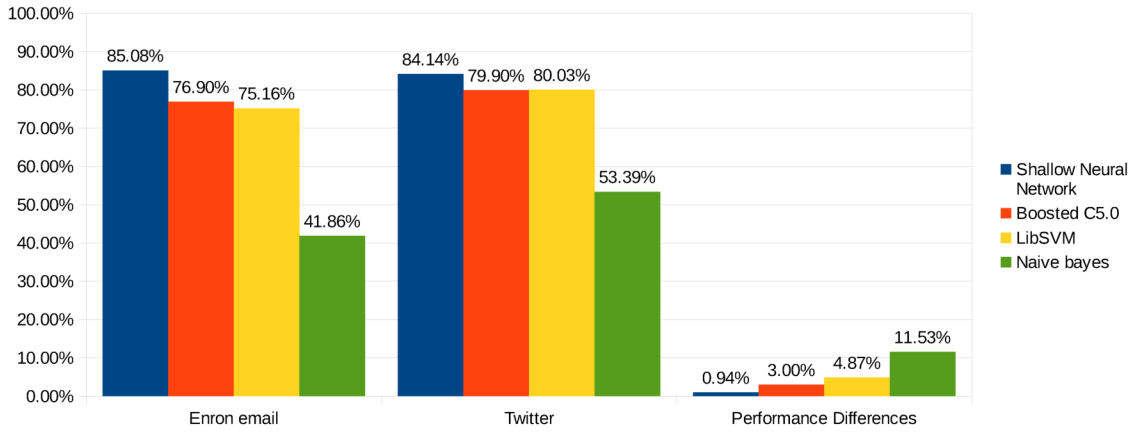

Figure 4.1: Summary of 10-fold cross validation results for 5 users

Fig. 4.1 illustrates 10-fold cross validation of the four algorithms on Enron email

and Twitter datasets as well as the difference in the accuracy between the two datasets for each algorithm. It is evident that the shallow neural network outperforms the other algorithms. Moreover, the performance variation of the proposed approach between two datasets is very small (only 0.94%) despite the fact that the data are totally different as shown in section 3.3 of Chapter 3. This shows that the proposed system is able to learn word embeddings - representation of the language attributes - from Reuters dataset and is able to generalize them to the Enron email and Twitter datasets to differentiate short texts written by different users.

4.1.2 Accuracy Evaluations on Ten Users

In order to evaluate the performance of the proposed system on more users, five more users are chosen from each dataset, to ten users on each dataset. The lists of five new users in Enron and Twitter datasets are listed in the Table 4.2 and 4.3, respectively. Note that the positions of the users in Enron company are also precious information for further digital forensic that showed in the next section, Section 4.2. The machine learning algorithms are re-run in the new set of 10 users to compare with the performance in the set of 5 users.

Table 4.2: Information of five new users from the Enron dataset

Username	**Position**
lenhart-m	*Employee*
rogers-b	*Employee*
dasovich-j	*Employee*
scott-s	*Trader*
arnold-j	*Vice President*

Fig. 4.2 reveals the 10-fold cross validation on the sets of ten users, and the information of performance differences between the two datasets are also included. It is apparent that artificial neural network still keeps its top 1 position among the four algorithms. Moreover, the performance difference of artificial neural network is only 0.40%, which makes it the most stable algorithm in term of accuracy.

In the results on the set of ten user, it can be noticed that the accuracy of LibSVM on is almost equal to artificial neural network (only 1.49% less) on the Twitter dataset.

Table 4.3: Information of five new users from the Twitter dataset

Username
Alexkeith53
CarrieLynn80
danlester43
PiperScott1949
RAYGARCIA71

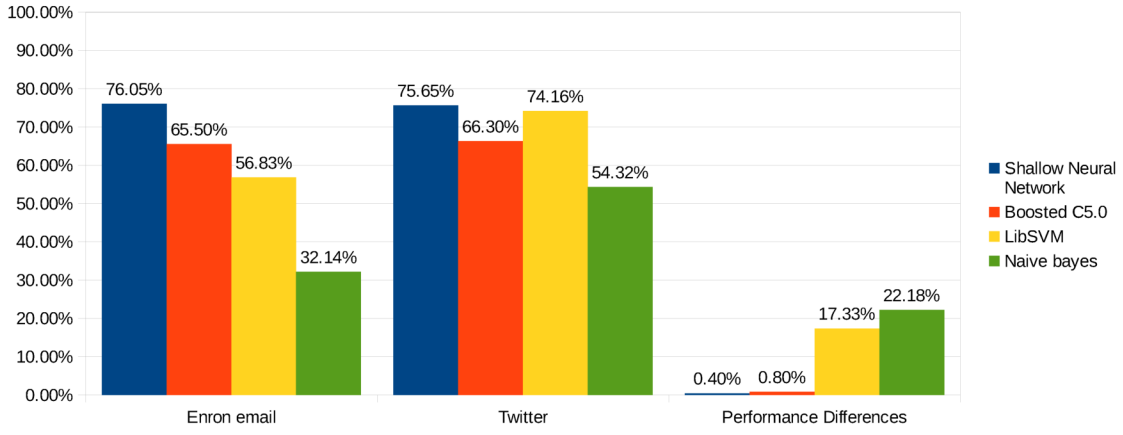

Figure 4.2: Summary of 10-fold cross validation results for 10 users

However, LibSVM's accuracy difference between the two datasets is high (17.33%) due to the lower accuracy on Enron dataset. Therefore, it can be concluded that LibSVM does not work as stable as artificial neural network. Meanwhile, C5.0 works stably (the accuracy difference is only 0.80%), but its accuracy on both datasets are approximately 10% lower than artificial neural network. Naïve bayes remains the worst algorithm in this evaluation.

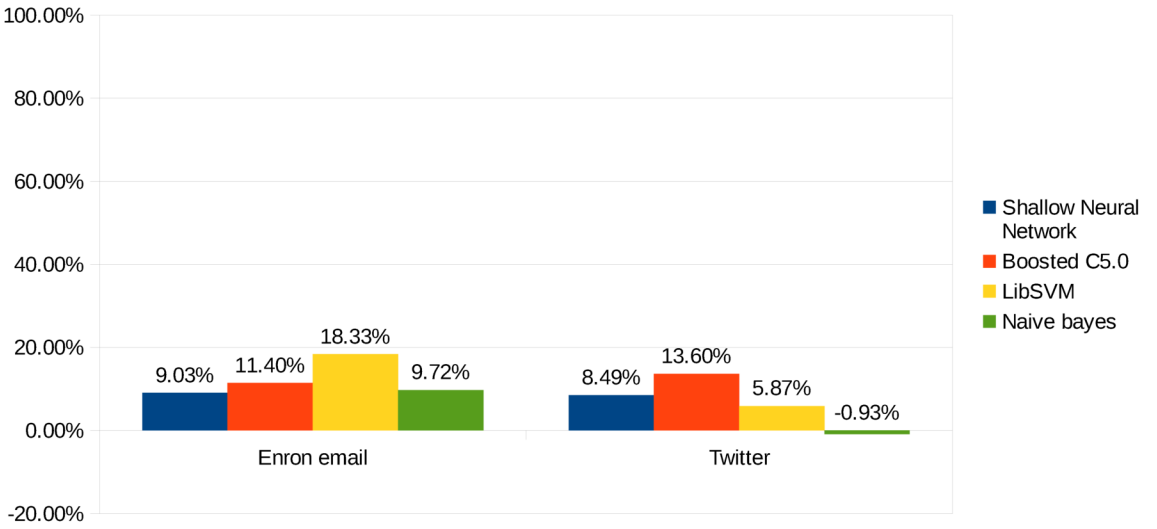

Figure 4.3: Summary of 10-fold cross validation performance drop

Performance drop when increase the number of users from five to ten (which is illustrated in Fig. 4.3) is an importance criteria to evaluate the machine learning algorithms. Artificial neural network continues to show its advantage in this evaluation, when its accuracy only drop around 9% in both datasets, while the number of users doubled from five to ten. Naïve bayes is an interesting case, its accuracy on Twitter dataset does not drop but increase by 0.93%. However, this does not mean the algorithm has better results on more users, since its instability is high as discussed earlier.

4.1.3 False Alarm Rate

False alarm rate or false positive rate [39] is employed in spam and impersonation detection systems [8] [9] [10] as an evaluation of the probability to generate false

alarms of the systems. Hence, false alarm rate is calculated on a per-class basic, which means we only consider "in-class" and "out-class" at any given class. Fig. 4.4 illustrates True Positive, True Negative, False Positive, and False Negative regions on a confusion matrix for a given class.

The false positive rate reflects the proportion of "out-class" samples that are erroneously classified as "in-class". False positive rate is calculated by dividing the number of "out-class" samples wrongly categorized as "in-class" (which is false positives) and the total number of actual "out-class" samples (which is the sum of true negatives and false positives) as stated in Eq. 4.1.

$$FPR = \frac{FP}{FP + TN} \tag{4.1}$$

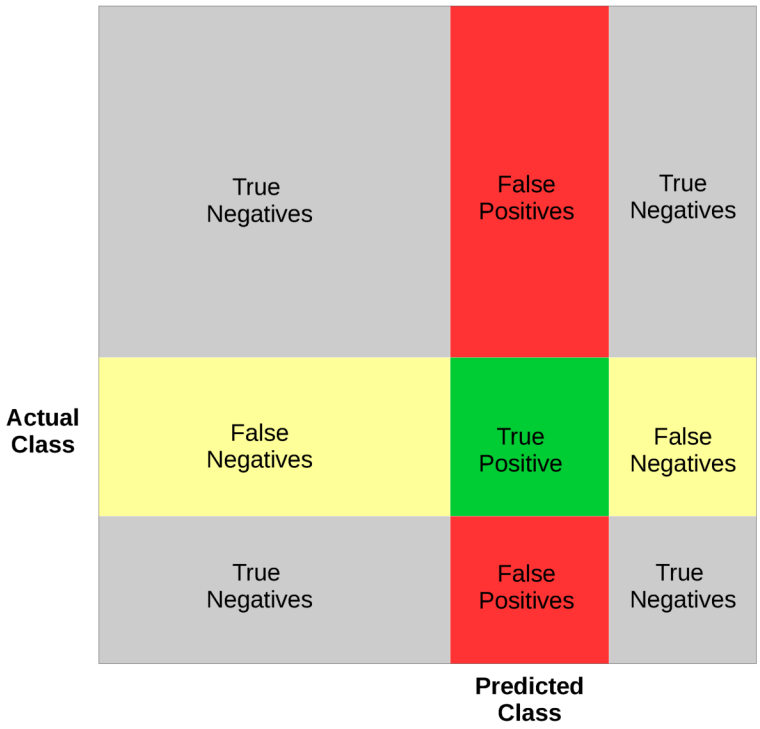

Figure 4.4: Confusion Matrix [40]

Table 4.4 shows the statistics of false positive rates on 10-fold cross validation of each algorithm. Confusion matrices in the 10 folds are added together and false positive rates are calculated from the output matrix. In this evaluation, artificial

neural network is still the best since it has the smallest mean and standard deviation of false positive rates (under the same dataset and number of users). Note that the means and standards deviation of 10-user groups are smaller than 5-user groups (within the same algorithm) since the true negative region in the confusion matrices expand quickly when the number of user is increased. Note that even though the False Positive Rates drop, the accuracy of all the evaluated algorithms also drop when increasing from 5 to 10 users. More evaluations need to be carried out in order to evaluate the increasing/decreasing trend of False Positive Rates when the number of users are changed.

Table 4.4: False Positive Rate Evaluation

Algorithm		**Enron Email Dataset**		**Twitter Dataset**	
		5 users	**10 users**	**5 users**	**10 users**
ANN[a]	FPR[b]	3.74%	2.72%	3.96%	2.67%
	SD[c]	2.13%	1.24%	1.61%	1.34%
C5.0 (Boosted)	FPR	5.78%	3.83%	5.02%	3.75%
	SD	2.64%	1.35%	1.34%	2.04%
libSVM	*FPR*	6.25%	4.83%	5.17%	3.0%
	SD	3.55%	5.42%	2.22%	2.05%
Naïve Bayes	FPR	14.54%	7.54%	11.65%	5.08%
	SD	15.62%	8.91%	7.96%	5.74%

[a]Artificial Neural Network
[b]Average of False Positive Rate of all users
[c]Standard Deviation of False Positive Rates.

4.1.4 Summary

In this section, there are 4 different criterias, including accuracy, difference of accuracy between the two dataset, accuracy drop, and false alarm ratio are evaluated on four different machine learning algorithms. These algorithms are trained and tested on the same short text features extracted from users' emails or tweets. All the evaluations show that artificial neural network outperfoms the other algorithms, and is a candidate for a potential effective solution against detecting compromised accounts or impersonation.

4.2 Further Digital Forensics

Beside the promising results showed in the previous section, this thesis also propose some forensic methods to assist the proposed system in detecting compromised accounts or impersonation. One forensic method is discussed in Section 3.4.2 of the Chapter 3, where the short texts of users are visualized and analyzed. This sections illustrated an higher level of forensic, that is looking at the writing style of different users and groups of users at the same time.

4.2.1 Word Cloud Representation Groups of Users

Word clouds are able to visualize the key words and their frequencies through the size of the keywords in the word cloud (i.e.,the bigger the more frequent). The text pre-processing steps are listed as following:

1. Combining texts of all users in the group.

2. Removing numbers and punctuation.

3. Stripping multiple white spaces and new lines.

The pre-processed texts are then passed through a counter to count the frequencies of words. Only the most frequent word are included in the word clouds, and in this thesis, the top 200 most frequent words are visualized.

The groups of users are determined by their positions/jobs in the Enron company. Note that this forensic is only carried out on Enron dataset due to the lack of job information in Tweeter dataset. The assumption is that users in a same job/position have similar writing style and can be distinguished from other groups, and this is analyzed further in the next section - Word Clouds Distances Visualization.

First of all, the word clouds represent Employee group (Fig. 4.5) and Trader group (Fig. 4.6) are similar. The reason is the top 200 frequent words of the two groups are similar. As an example for the similarity, the top 10 most frequent word of the Employee group are listed in the Table 4.5. Due to the similarity, the two groups are combined and from a group of Employees and Traders. The word cloud of the new group is illustrated in the Fig. 4.7.

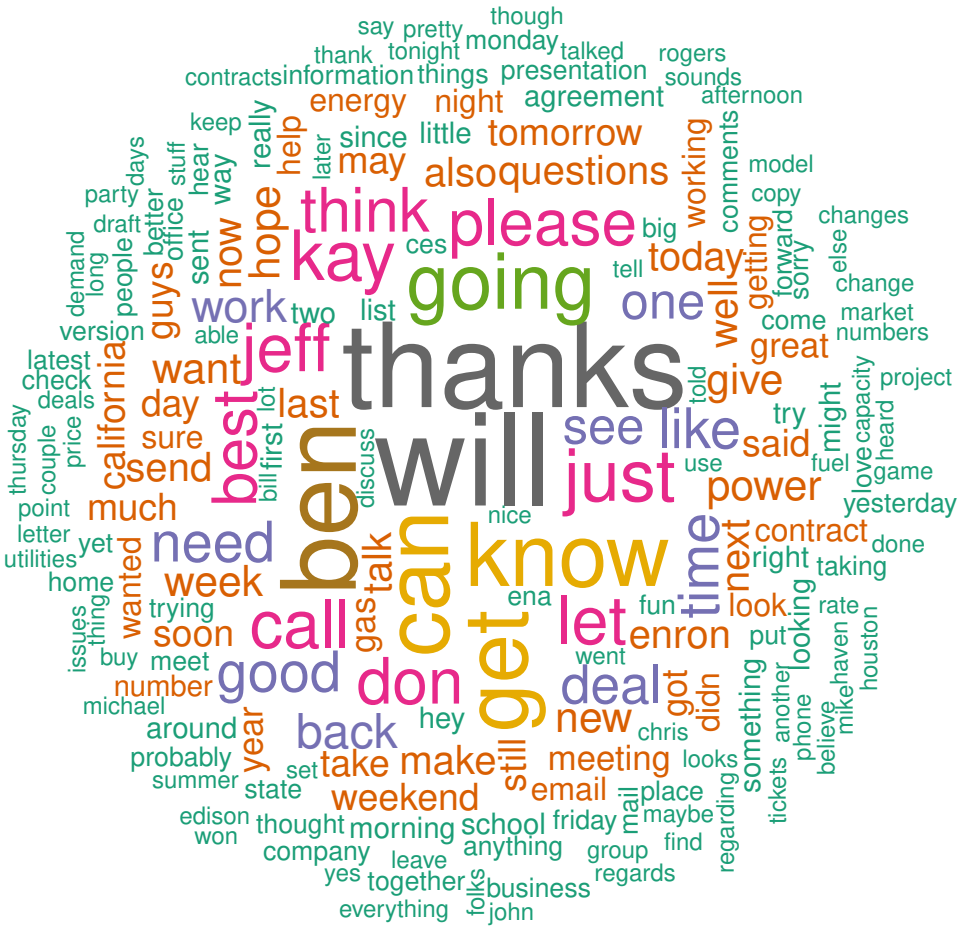

Figure 4.5: Word cloud of Employee group in Enron email dataset

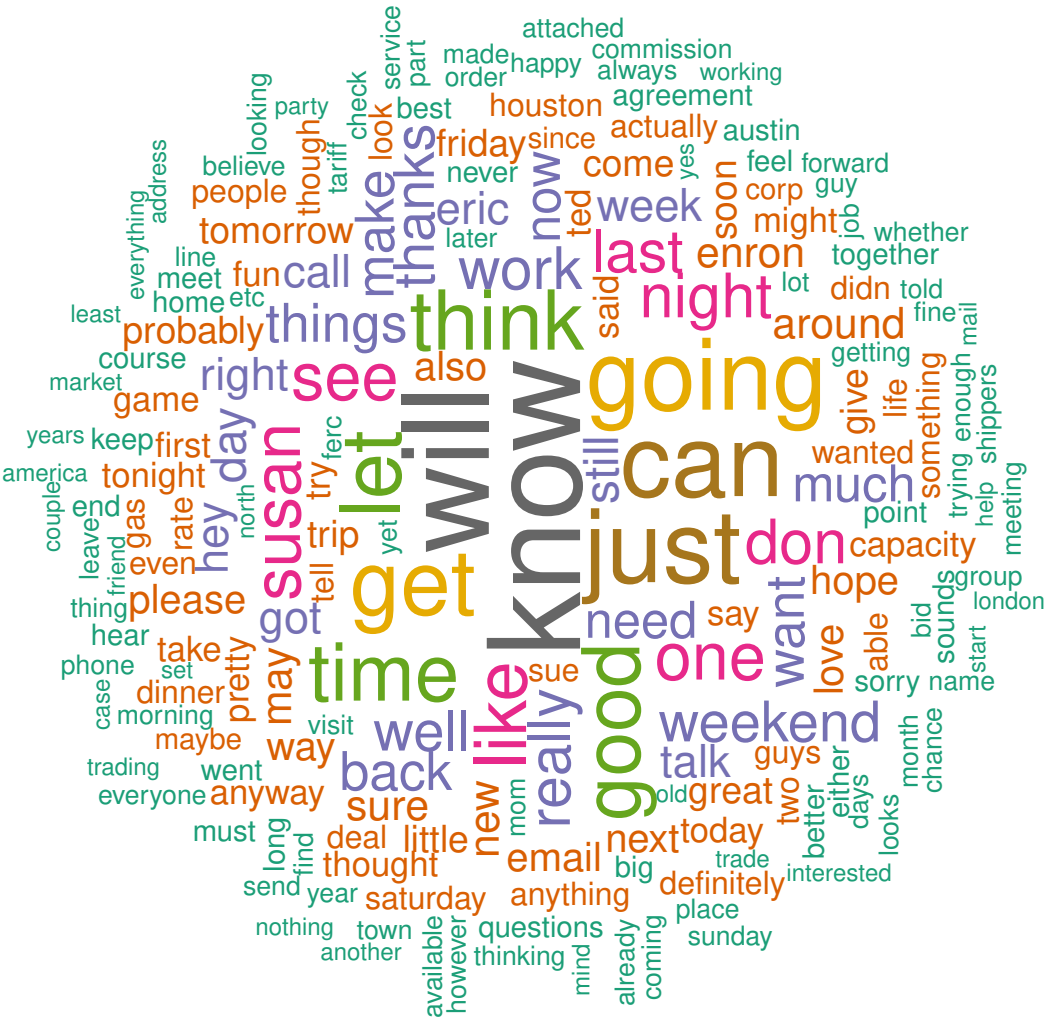

Figure 4.6: Word cloud of Trader group in Enron email dataset

Figure 4.7: Word cloud of Emplyee and Trader groups in Enron email dataset

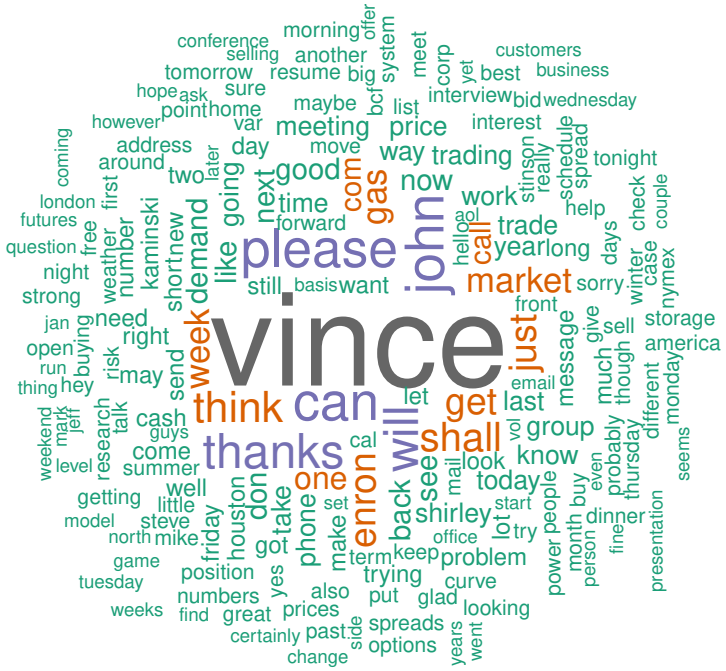

Figure 4.8: Word cloud of Manager group in Enron email dataset

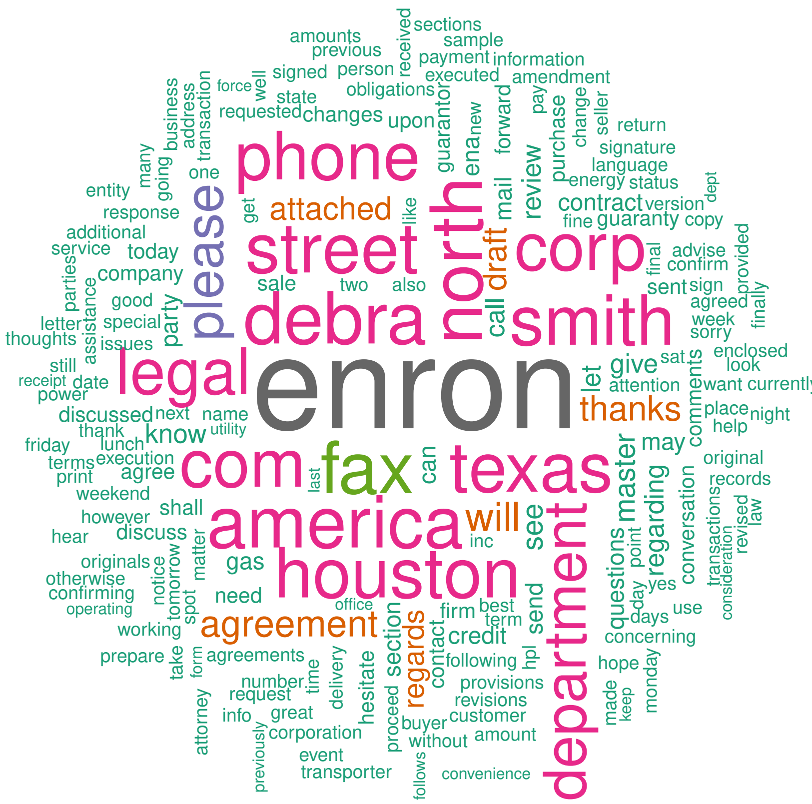

Figure 4.9: Word cloud of Legal Department in Enron email dataset

Table 4.5: Top ten most frequent words in Employee and Trader group

Employee group	**Trader group**
will	know
thanks	will
can	can
know	thanks
get	get
going	going
just	just
let	let
call	think
please	time

The Fig. 4.8 and 4.9 are the word clouds of the manager and legal department groups, respectively. It could be deduced from the Fig. 4.8 that mangers often write short emails with less repeated words than the other groups; Meanwhile, the word cloud of legal department contain some keywords that are related to their job, such as "agreement" or "legal". This section shows that simple word clouds could effectively strengthen the assumption that there are the differences in the writing style between users and group of users.

4.2.2 Word Clouds Distances Visualization

The comments in the previous section are quite qualitative. Hence, a more quantitative parameter need to be employed in order to form a good forensic tool. The visualization showed in Fig. 4.10 is an effort to "quantifying" the distances between the word clouds, and we can see which group's or user's word clouds are nearer to each other.

The figure is obtained by using t-SNE, the visualization tool which is used to visualize word embeddings in Section 3.4.1 of the Chapter 3. The input features for the t-SNE are similar to the short text feature and are extracted by performing the following steps:

1. Combining emails of the users (or all emails of the user in the group).

2. Removing numbers and punctuation and stripping multiple white spaces and

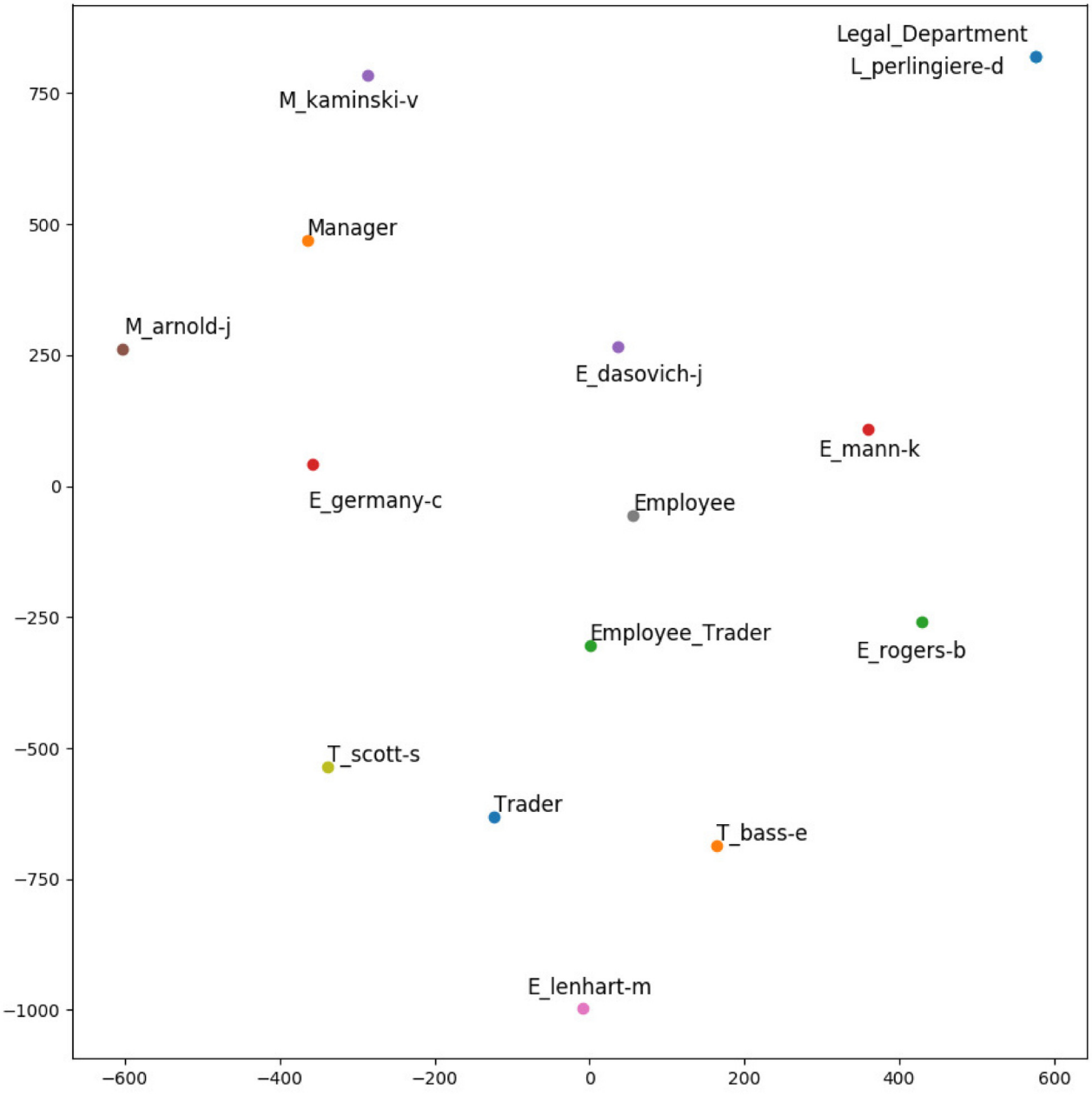

Figure 4.10: Word clouds distances visualization

new lines.

3. Creating word clouds (with frequencies of words) for all users and groups of users.

4. Extracting all the words and their frequencies in each word cloud.

5. Forming the "short text of word cloud" by repeat each extracted word with its frequency.

6. Extracting short text features (which is consider as the features extracted from each word cloud).

In the picture, 5 groups and 10 users are visualized, and the groups include *Manager, Legal_Department, Employee, Trader, and Employee_Trader*. Users have prefix of the groups that they belong to and follow by user's name. The prefixes including: *M_ : Manager, L_: Legal department, E_: Employee, T_: Trader*.

It is evident that the 3 groups: Employee, Trader, and Employee_Trader are near to each other and are surrounding by all the users who were Employee and Trader. This supports the hypothesis in the analysis of word cloud's keywords in the previous section. The characteristic repeats in Manager group and Manager users. The Legal_department group only has 1 user, so the group and user's word cloud are identical.

Chapter 5

Conclusion and Future Work

This thesis builds a machine learning based language model that aims to identify compromised users. The proposed system is simple and able to work with different types of data while still remain stable and good performance. This chapter is a conclusion of this work and future work are also listed.

5.1 Conclusion

In this research, the usage of a shallow neural network is explored for identifying compromised users. To this end, an authorship attribution approach is employed on discovering the writing styles of users on short texts such as emails or tweets. Then, the discovered writing styles is used to identify/differentiate the users from each other.

The proposed system potentially capable of differentiating a compromised account where the attacker imitates to be the legitimate user. In the design of the proposed approach, users' and attackers' short texts can belong to various languages or datasets, but they must in the same language. In other words, the proposed model could be applied in various languages. However, this study only performed on English.

The proposed system is evaluated on two different datasets, namely Enron and Twitter, against three different classifiers: Naïve Bayes, C5.0 and SVM. Further more, the proposed system is able to:

- Work with arbitrary length of short texts.
- Learn the language model in un-supervised manner.
- Train the decision maker in supervised manner.
- Assist digital forensic.

The results show that the shallow neural network outperforms the other classifiers on these data in all the evaluated criterias, including accuracy, performance difference,

performance drop, and false alarm ratio (See Chapter 4). Furthermore, the proposed approach is able to learn the language model on one dataset namely RCV1, and able to generalize this model without modification to Enron and Twitter data with approximately 85% accuracy on the set of five users and 76% on the set of ten users. In other words, the low-level word embeddings, modelled from RCV1, is used to form high-level features effectively and able to identify the users in the Enron email and Twitter datasets.

This thesis also investigate users' and group of users' writing style (See Section 4.2 of Chapter 4), and the out come results strengthen the assumption that each user has his or her own writing style and the same job/position have similar writing style that can be distinguished from other groups.

In terms of forensic analysis this indicates a powerful tool which can be trained in the lab but could easily generalize to the wild. Some forensic applications that are demonstrated in this work include:

- Word cloud visualization that is employed to get an insight into the datasets' characteristic.
- Visualization of users' short text based on the text features proposed in this thesis. The visualization give an insight into the different writing styles of users.
- Word cloud visualization that is employed to get an insight into the difference groups' writing styles.
- Visualization of "distance" between the word cloud of users and groups to get an insight into the "relation" between users' and groups' writing styles.

5.2 Future Works

The evaluations show that the proposed system generates very promising results. The results demonstrate the overall consistency of the proposed model compared to the other learning systems. In future studies, the proposed system can be developed to become a powerful forensic and security tool. In order to do so, some important works need to be done are:

- Improve the accuracy on bigger number of users.
- Evaluate more datasets and machine learning algorithms.
- Work with different feature extracting methods.
- Work with previously unseen users.
- Embed digital forensic tools in to the system.

Bibliography

[1] Michelle Alvarez, Nicholas Bradley, Pamela Cobb, Scott Craig, Ralf Iffert, Limor Kessem, Jason Kravitz, Dave McMillen, and Scott Moore. Ibm x-force threat intelligence index 2017 the year of the mega breach. *IBM Security,(March)*, pages 1–30, 2017.

[2] Cleary Gillian, Corpin Mayee, Cox Orla, Lau Hon, Nahorney Benjamin, O'Brien Dick, O'Gorman Brigid, Power John-Paul, Wallace Scott, Wood Paul, Wueest Candid, et al. Symantec internet security threat report. *Volume 23*, 2018.

[3] https://www.emarsys.com/en/resources/blog/a-brief-history-of-spam-filtering-and-deliverability-gunter-haselberger/.

[4] https://techcrunch.com/2017/06/27/facebook-2-billion-users/.

[5] Steven HH Ding, Benjamin Fung, and Mourad Debbabi. A visualizable evidence-driven approach for authorship attribution. *ACM Transactions on Information and System Security (TISSEC)*, 17(3):12, 2015.

[6] https://www.nbcnews.com/tech/security/hbo-investigating-hack-its-twitter-accounts-n793391.

[7] https://techcrunch.com/2017/03/15/twitter-counter-hacked/.

[8] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 35–47. ACM, 2010.

[9] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @ spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 27–37. ACM, 2010.

[10] Brett Stone-Gross, Thorsten Holz, Gianluca Stringhini, and Giovanni Vigna. The underground economy of spam: A botmaster's perspective of coordinating large-scale spam campaigns. *LEET*, 11:4–4, 2011.

[11] Kurt Thomas, Chris Grier, and Vern Paxson. Adapting social spam infrastructure for political censorship. In *LEET*, 2012.

[12] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *Proceedings of the 26th annual computer security applications conference*, pages 1–9. ACM, 2010.

[13] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Compa: Detecting compromised accounts on social networks. In *NDSS*, 2013.

[14] Kurt Thomas, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse. In *USENIX Security Symposium*, pages 195–210, 2013.

[15] Xianghan Zheng, Zhipeng Zeng, Zheyi Chen, Yuanlong Yu, and Chunming Rong. Detecting spammers on social networks. *Neurocomputing*, 159:27–34, 2015.

[16] Abdur Rahman MA Basher and Benjamin CM Fung. Analyzing topics and authors in chat logs for crime investigation. *Knowledge and information systems*, 39(2):351–381, 2014.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[18] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3 (Feb):1137–1155, 2003.

[19] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[20] George Bebis and Michael Georgiopoulos. Feed-forward neural networks. *IEEE Potentials*, 13(4):27–31, 1994.

[21] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436, 2015.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

[25] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2 (3):27, 2011.

[26] Larry M Manevitz and Malik Yousef. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154, 2001.

[27] https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html.

[28] https://www.rulequest.com/see5-unix.html.

[29] http://scikit-learn.org/stable/modules/tree.html.

[30] http://www.statsoft.com/textbook/naive-bayes-classifier, .

[31] http://scikit-learn.org/stable/modules/naive_bayes.html, .

[32] Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung Hyon Myaeng. Some effective techniques for naive bayes text classification. *IEEE transactions on knowledge and data engineering*, 18(11):1457–1466, 2006.

[33] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

[34] https://www.cs.cmu.edu/~enron/.

[35] Zhou Yilu, Alsarkal Yaqoub, and Zhang Nan. Linking virtual and real-world identities twitter dataset. http://www.azsecure-data.org/, 2016.

[36] Yaqoub Alsarkal, Nan Zhang, and Yilu Zhou. Linking virtual and real-world identities. In *Intelligence and Security Informatics (ISI), 2015 IEEE International Conference on*, pages 49–54. IEEE, 2015.

[37] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[38] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. Api design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013.

[39] Nathalie Japkowicz and Mohak Shah. *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.

[40] http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html.